ॐ

OM

9574234622

Sujal Patel (+918121564132)

PM 1(B)

ECE

Digital Circuits

Dr. Chakrapani.

Yerramsetty Chakri @ gmail.com.

Best of luck

✩ Multi Vibrators using logic gates.

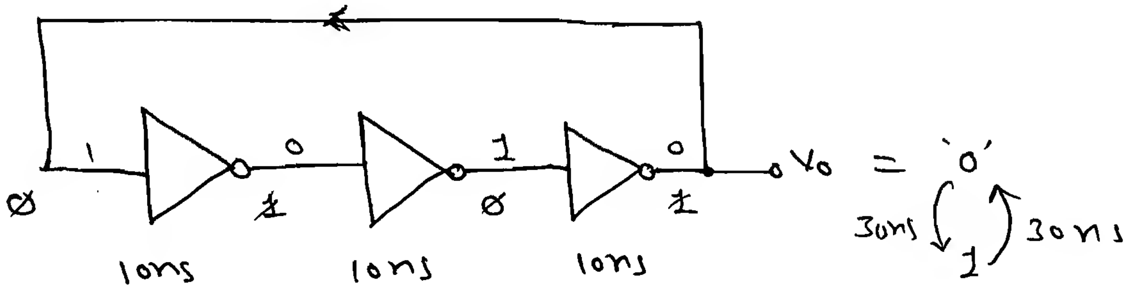
Applications

① Astable Multivibrators → Square wave generator
(Free running mv)

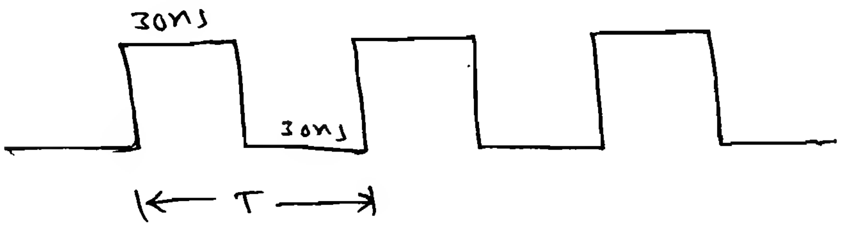
② Monostable multivibrators → { (a) Pulse generator
(One shot mv) (b) Pulse stretcher.

③ Bistable multivibrators → 1-Bit memory.

* Astable Multivibrators Using logic gate.



'0', '1' = Unstable States.



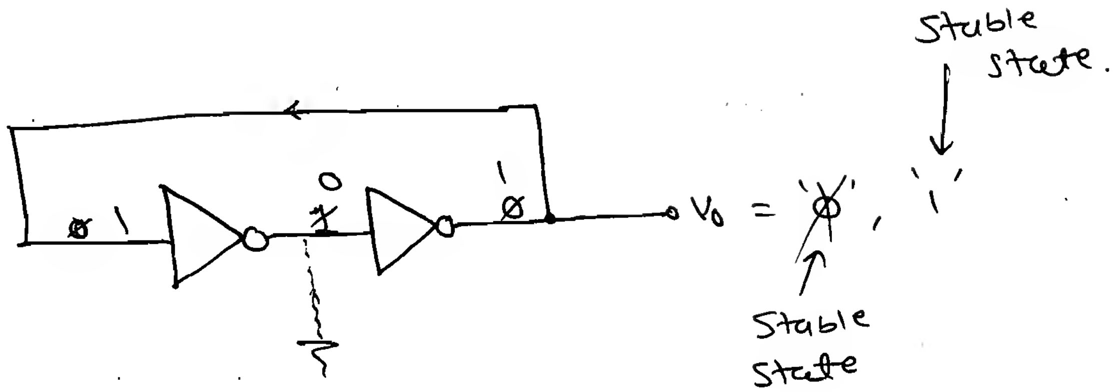
$\therefore \boxed{T = 60\ ns}$

→ Time period of Square wave

$$\boxed{T = 2 \times [\text{sum of Propagation Delays of all Inverters}]}$$

$\therefore \boxed{f = 1/T}$

* Bistable Multivibrators.

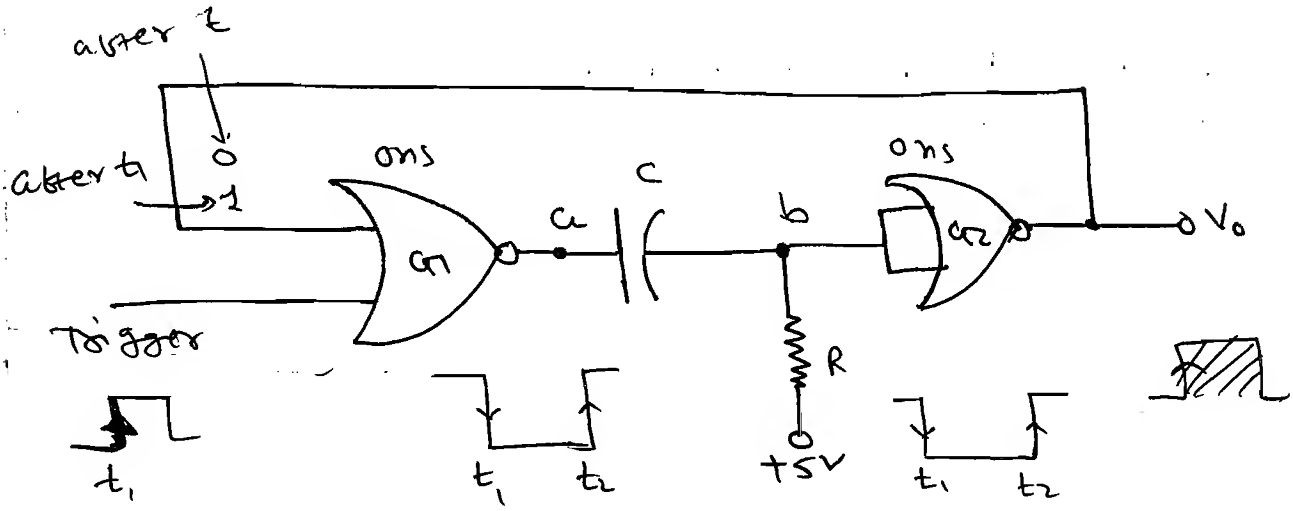


NOTE:

→ If odd no. of Inverter then ~~was~~ Astable MV.

→ If even no. of Inverter then Bistable MV.

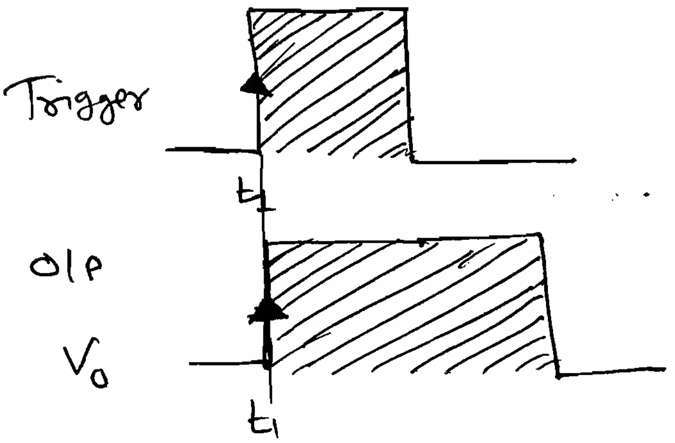
* Monostable Multivibrator



→ After T1, the feedback came at gate-① input is equal to '1' which means the Va = 0. Capacitor charges to +5V with time constant RC.

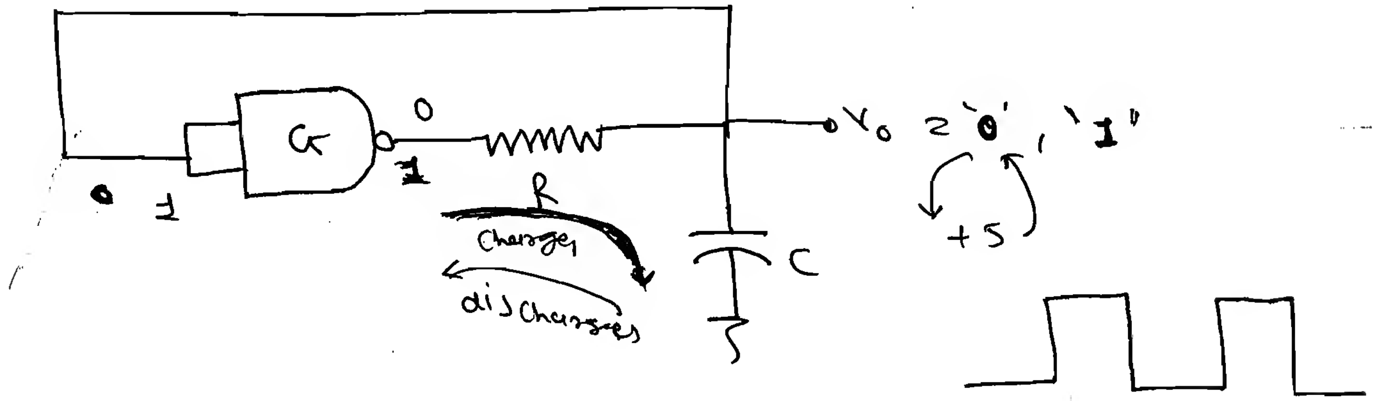
→ Once capacitor charges to +5 Volts. Vb = logic '1' ⇒ $\boxed{V_o = 0}$ and $\boxed{V_a = 1}$ and hence the capacitor discharges.

* Pulse Stretching.



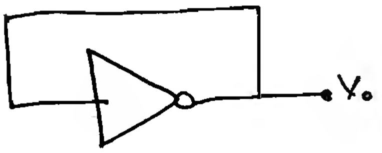
Ex- Determine the following multivibrators:

a)



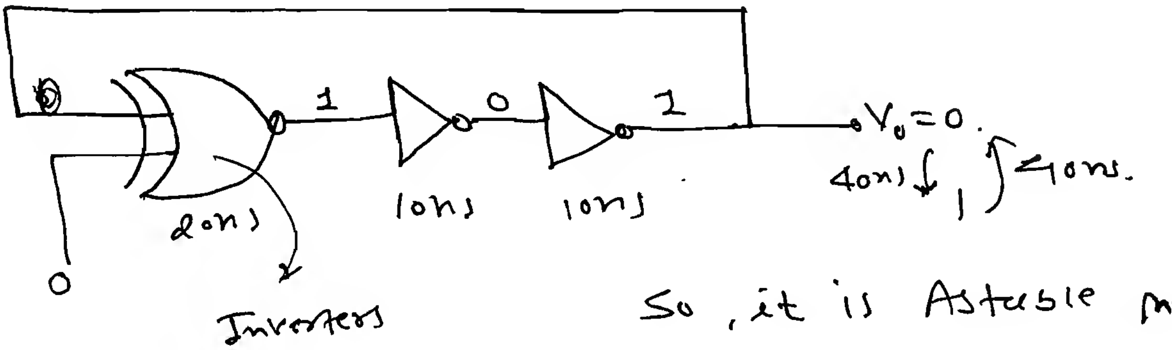
→ So, it is Astable mv.

b)



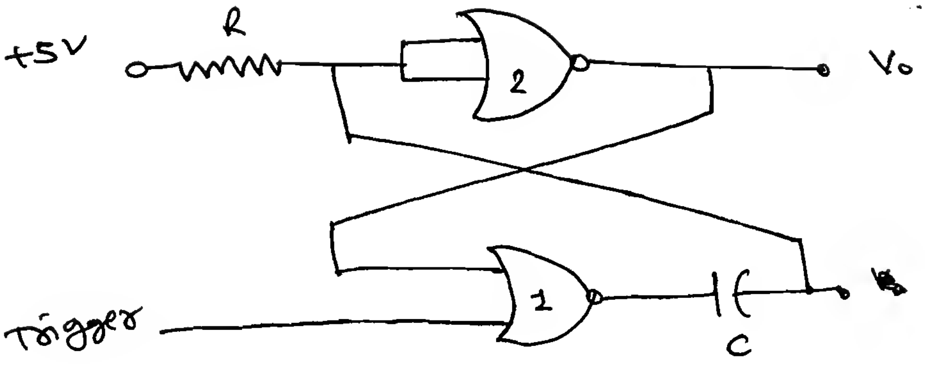
→ Astable mv.

c)

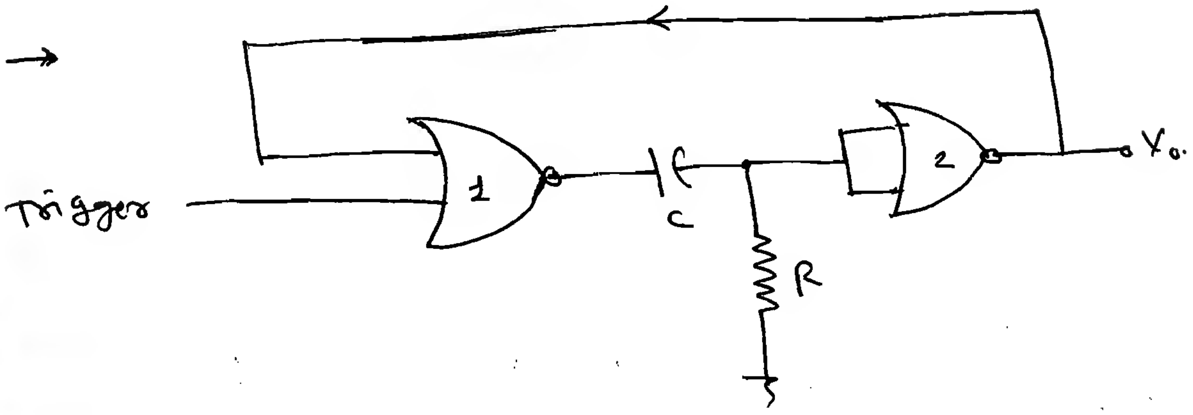


So, it is Astable mv

④



|||

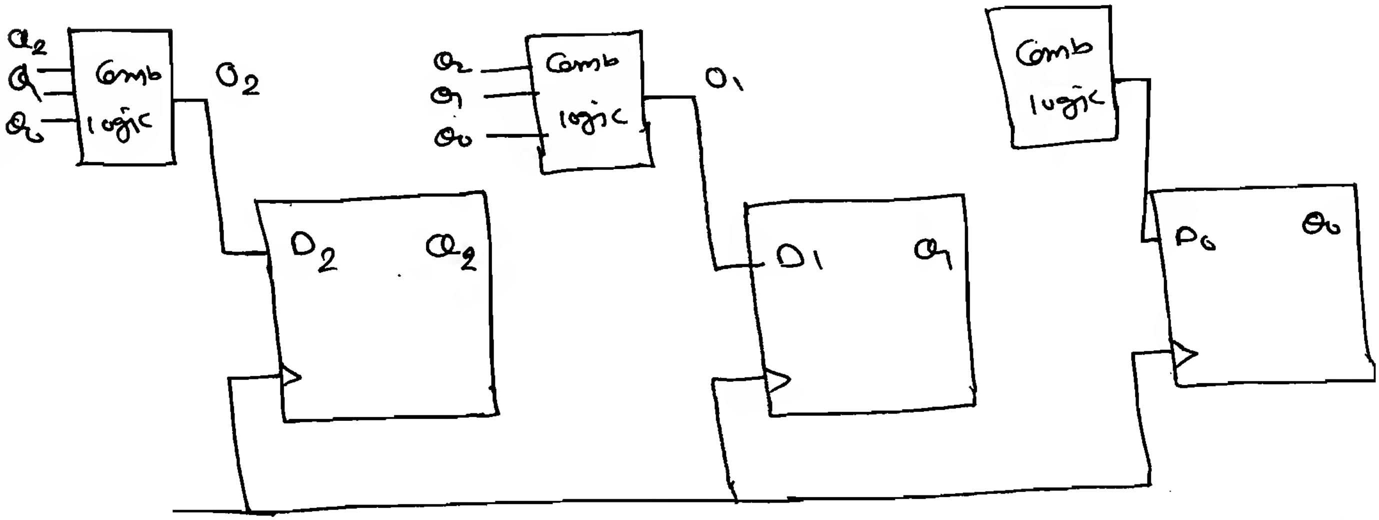


→ Now it is look like monostable mv.

So, given ckt is monostable mv.

Ex- In the following circuits determine the values of O_2, O_1, O_0 = ? if it is counting the following sequence 101, 010, 110

→



→ This is nothing but design a sync counter using D-FF which counts through the states 5, 2, 6, 5, ...

→ at time of design use excitation table

→ at time of analysis use truth table.

→

	P.S.			N.S			FF Inputs		
	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	D_2	D_1	D_0
5	1	0	1	0	1	0	0	1	0
2	0	1	0	1	1	0	1	1	0
6	1	1	0	1	0	1	1	0	1

→ $\boxed{D_2 = Q_1}$

$D_2 = \Sigma m(2,5) + d(0,3,1,4,7)$

→

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0	X	X	X	1
1	X	1	X	0

∴ $D_1 = \bar{Q}_2 + Q_0$.

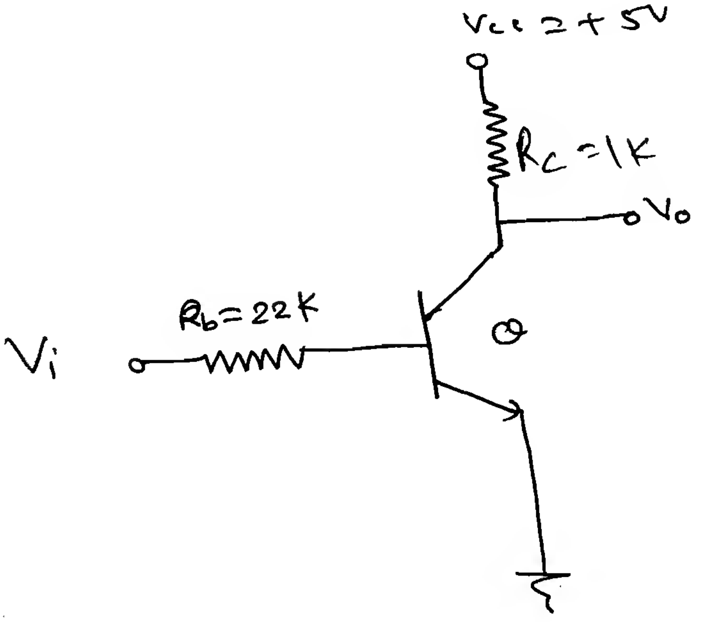
$D_1 = \Sigma m(6) + d(0,1,3,4,7)$

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0	x	x	X	0
1	X	0	X	1

$D_1 = \bar{Q}_0 Q_2$ (or) $Q_2 . Q_1$.

☆ Logic Families:

* Transistor as an Inverter:



$V_{BE, cutoff} = 0.8V$

$V_{BE, active} = 0.7V$

$V_{BE, sat} = 0.8V$

(i) $V_i = 0V$; Q is OFF ⇒ $V_o = +5V$.

(ii) $V_i = +5V$; Q is ON ⇒ $V_o = V_{CE, sat} = 0.2$.

→ $I_B \geq I_{B, min}$. where $I_{B,min} = \frac{I_{c, sat}}{h_{FE}}$.

$\therefore \; I_B \geq \frac{I_{c, sat}}{h_{FE}}$.

$\therefore \; \boxed{h_{FE} I_B \geq I_{c, sat}}$ → For Q to be in Saturation.

→ $I_{c, sat} = \frac{V_{cc} - V_{CE, sat}}{R_c} = \frac{5 - 0.2}{1K} = 4.8 \text{ mA}.$

$\therefore \; I_B = \frac{V_i - V_{BE, sat}}{R_b} = \frac{5 - 0.8}{22} = \frac{4.2}{22} = 0.19 \text{ mA}$

$\therefore \; h_{FE} I_B = (20)(0.19) = \boxed{3.8 \text{ mA}.}$

* Q is not in Saturation, because,

$\boxed{h_{FE} I_B < I_{c, sat}}$

NOTE: In the above circuits Ib $h_{FE} = 30$ then

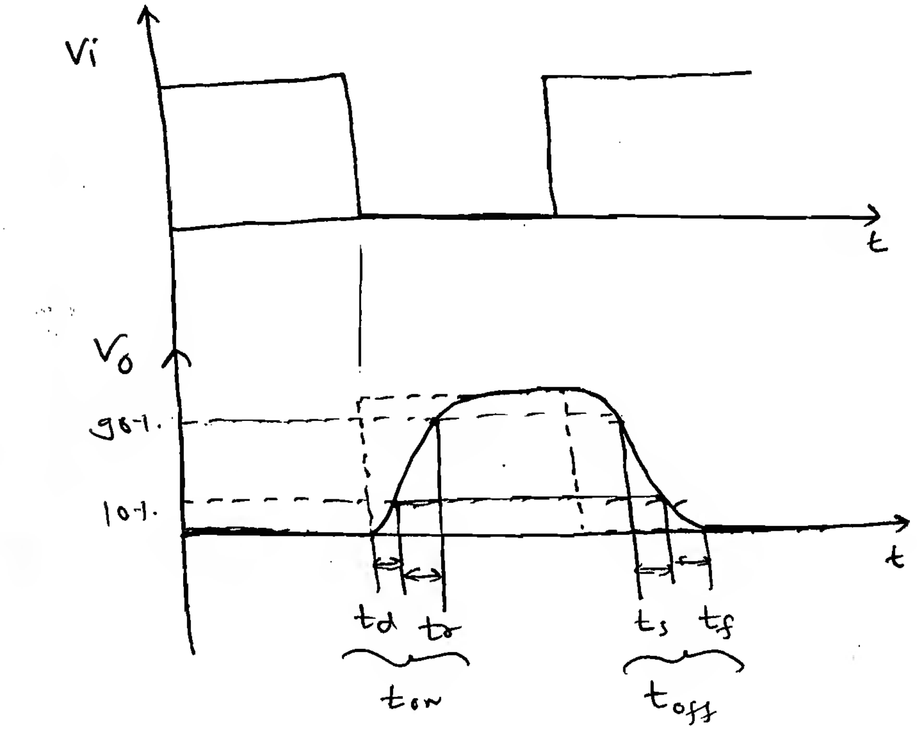
then $h_{FE} I_B = (30)(0.19) = 5.7mA.$

$\therefore \boxed{I_{C\,sat} = 4.8mA.}$

So, Transistor is in saturation, because

$h_{FE} I_B > I_{C\,sat}.$

* Transistor switching time:



t_d = delay time.

t_r = Rise time.

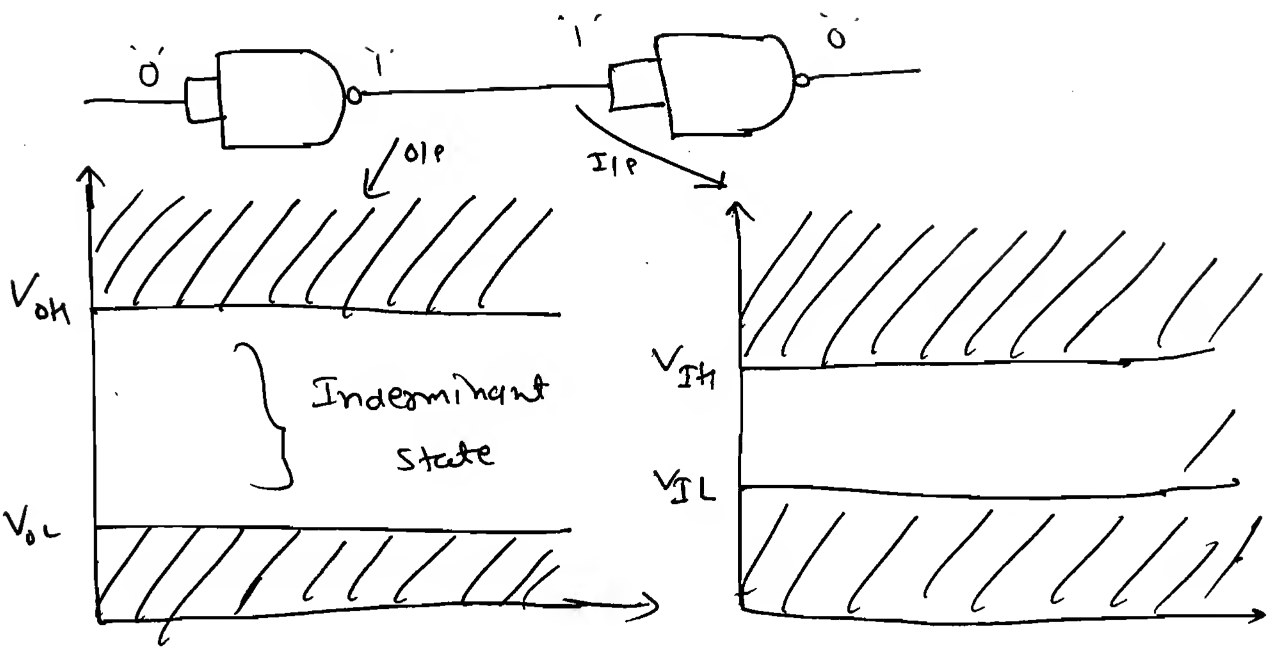
t_s = Storage time.

t_f = fall time.

→ $t_{ON} = t_d + t_r$.

$t_{OFF} = t_s + t_f$.

* Parameter of Logic gates:



$$\boxed{V_{OH} > V_{IH} > V_{IL} > V_{OL}.}$$

2) Noise margin:

→ It is the amount of Noise that can be allowed without disturbing the normal operation of logic gates.

✓ $NM_H = V_{OH} - V_{IH} = 2.4 - 2.0 = \boxed{0.4V}$ ↑ (-ve noise)

$NM_L = V_{IL} - V_{OL} = 0.9 - 0.4 = \boxed{0.5V} \rightarrow \downarrow$ (+ve noise)

$\therefore N.M. = \min(NM_H, NM_L).$

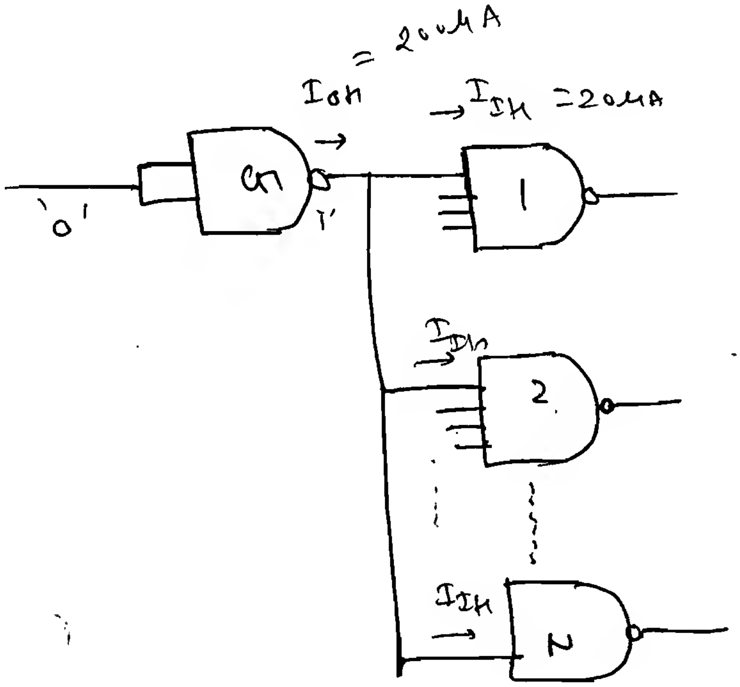
$= \min(0.5, 0.4).$

$\therefore \boxed{NM = 0.4V}$

③ Fanout:

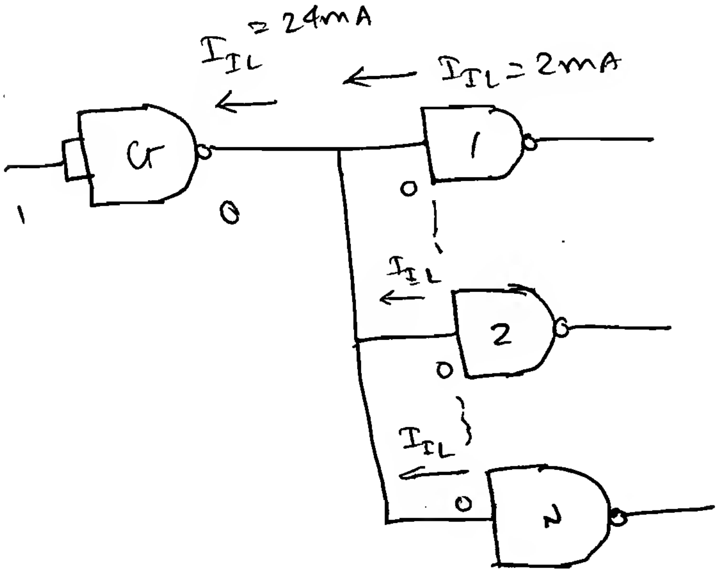
→ It is the no. of standard load o/p of gate can drive without impairing its normal operation. The standard load is input current of a logic gate which belongs to the the family of driving gate.

(a)



$$\text{Fan out}_H = \frac{I_{OH}}{I_{IH}} = 10.$$

(b)



$$\text{Fanout}_L = \frac{I_{OL}}{I_{IL}}$$

$$= \frac{24}{2}$$

$$\text{Fanout}_L = 12$$

→ Fanout = min (Fanout_H , Fanout_L).

= min (10, 12)

∴ $\boxed{\text{Fanout} = 10}$

④ FoM (figure of merit).

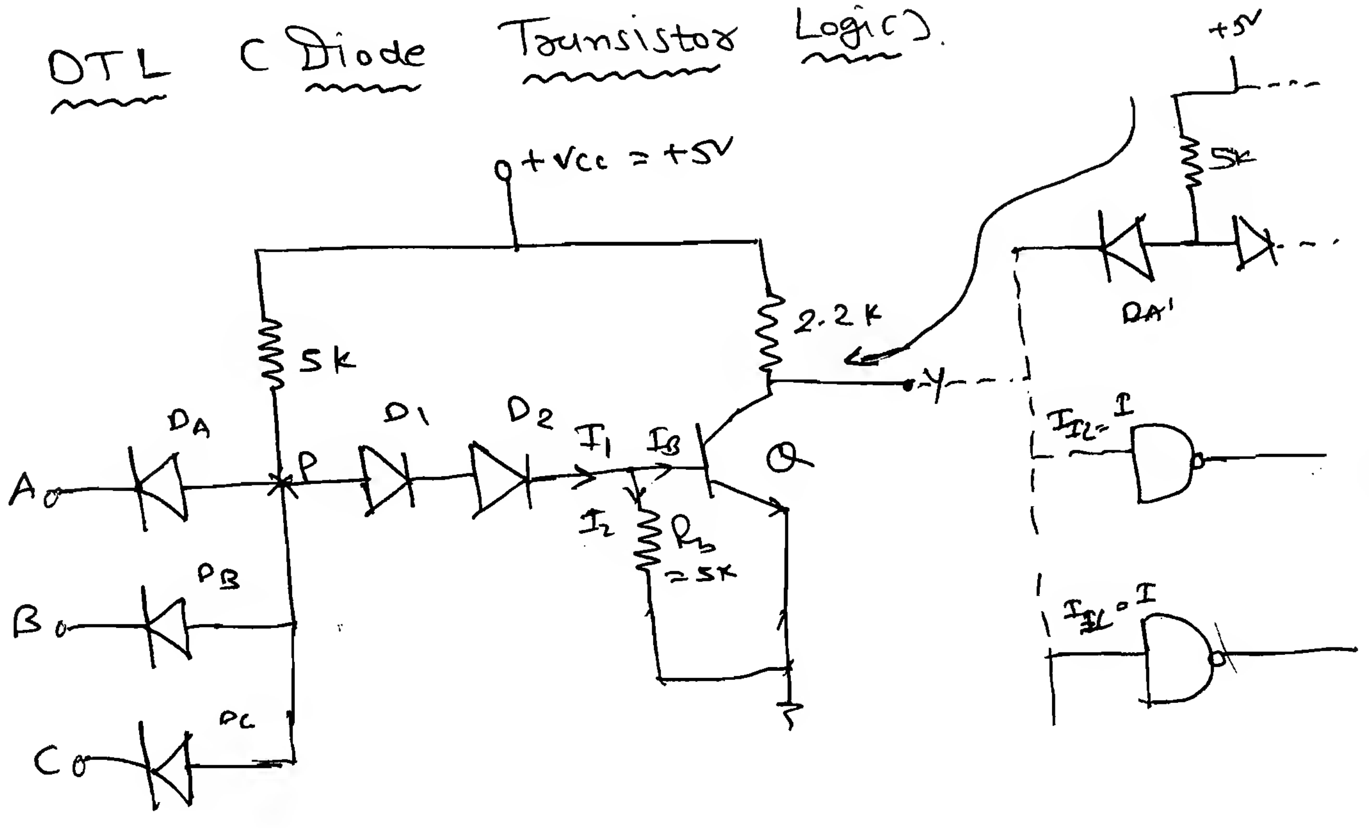
→ Prop. delay (ns) × power dissipiation (mW)

=> picoJoules (PJ).

→ It is used to compare two gates in terms of its performance.

→ ① DTL
② modified DTL
③ TTL
④ ECL.

① DTL (Diode Transistor Logic).



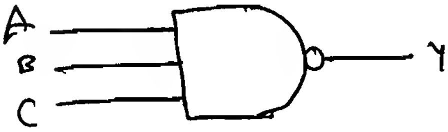
A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

$\rightarrow$ D_A, D_B, D_C are ON
D_1, D_2, Q are OFF $\Rightarrow Y = +5V.$

$\rightarrow$ $D_1, D_2,$ Q are OFF $\Rightarrow Y = +5V$

$\rightarrow$ D_A, D_B, D_C are OFF
D_1, D_2 Q are ON,
$Y = V_{CE,sat} = 0.2\,V.$

So, it is NAND gate.



$Y = \overline{A \cdot B \cdot C}.$

$\rightarrow$(i) when Q is on the Voltage at P

$V_P = 0.7 + 0.7 + 0.8 = \boxed{2.2\ V.}$

(ii) when Q is OFF $\Rightarrow V_P = 0.7 + 0.2 = \boxed{0.9V}$

$\rightarrow$ For 'Q' to be in saturation.

$h_{FE} I_B \geq I_{C,sat} + N.I.$

$I_{C,sat} = \dfrac{5 - V_{CE,sat}}{2.2} = \dfrac{4.8}{2.2} = \boxed{2.18\ mA.}$

Stand. Load I = ?

$$I = \frac{5 - V_A' - V_{CE_1 sat}}{5} = \frac{5 - 0.7 - 0.2}{5}.$$

$$\therefore \quad I = \frac{4.1}{5} = \boxed{0.82\,mA}$$

$$\rightarrow \quad I_1 = \frac{5 - V_P}{5} = \frac{5 - 2.2}{5}.$$

$$\therefore \quad I_1 = \frac{2.8}{5} = \boxed{0.58\,mA}$$

$$\rightarrow \quad I_2 = \frac{V_{BE,sat}}{R_b} = \frac{0.8}{5} = \boxed{0.16\,mA}$$

$$\therefore \quad I_B = I_1 - I_2 = 0.58 - 0.16$$

$$\therefore \quad \boxed{I_B = 0.4\,mA}$$

Let, $h_{FE} = 30.$

$$\therefore \quad 30 \times 0.4m \geq 2.18m \cancel{\times N} + N.(0.82)m$$

$$\therefore \quad 12 \cancel{+2} \geq 2.18 + (0.82 \times N)$$

$$\therefore \quad 9.82 \geq 0.82 \times N$$

$$\therefore \quad N \leq \frac{9.82}{0.82}.$$

$$\therefore \quad N = 11.97$$

$$\therefore \quad \boxed{N = 11}$$

$$\therefore \quad \boxed{\text{Fanout} = 11}$$

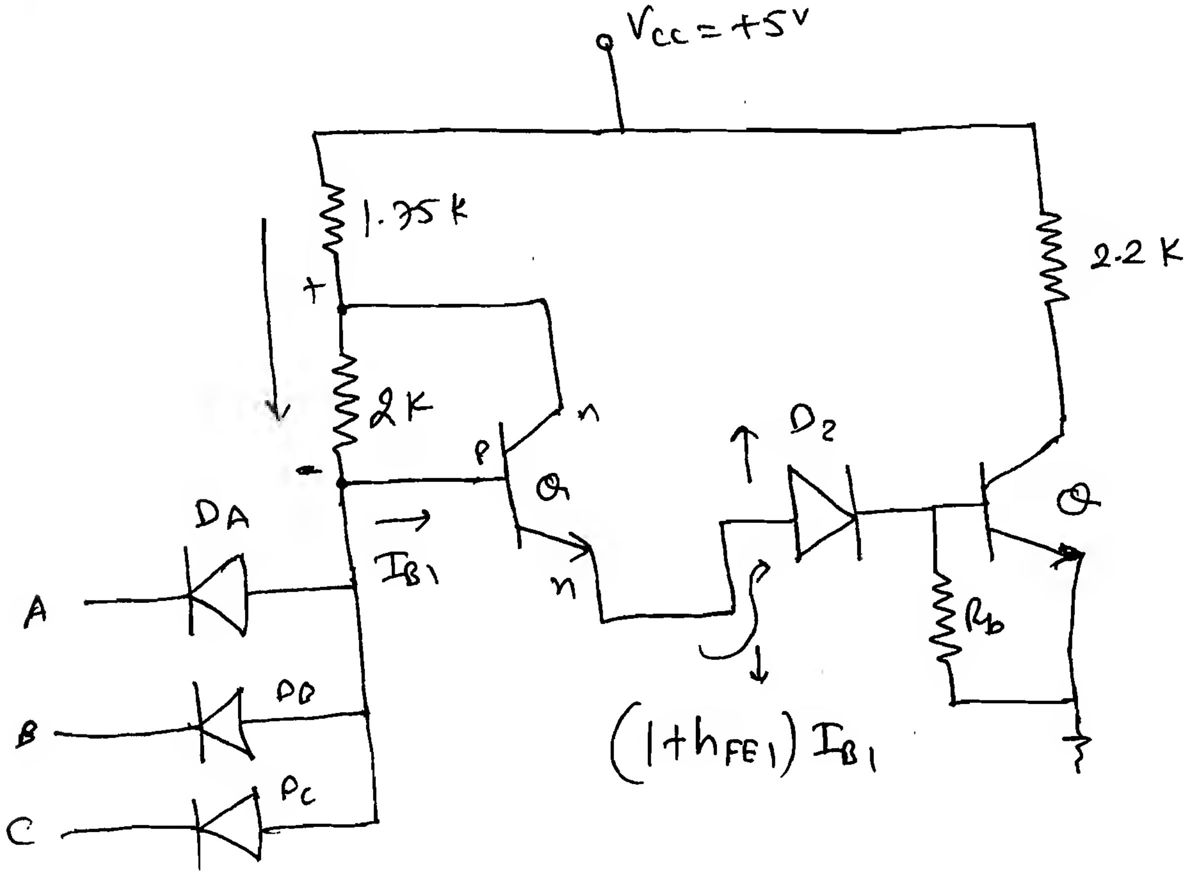
→ In OTL two Diodes D_1 & D_2 are used to increase the noise margin of the logic gates.

→ ~~At~~ In OTL, the R_b resistor is used to reduce the Storage time of the transistor which in turn reduces the switch off time of the transistor.

② Modified OTL Gate:

→ In modified DTL the Diode D_1 is replaced by a transistor in active region which increases the base current of Q. hence the fanout of the Logic increases

→



→ 'Q' is in Active region.

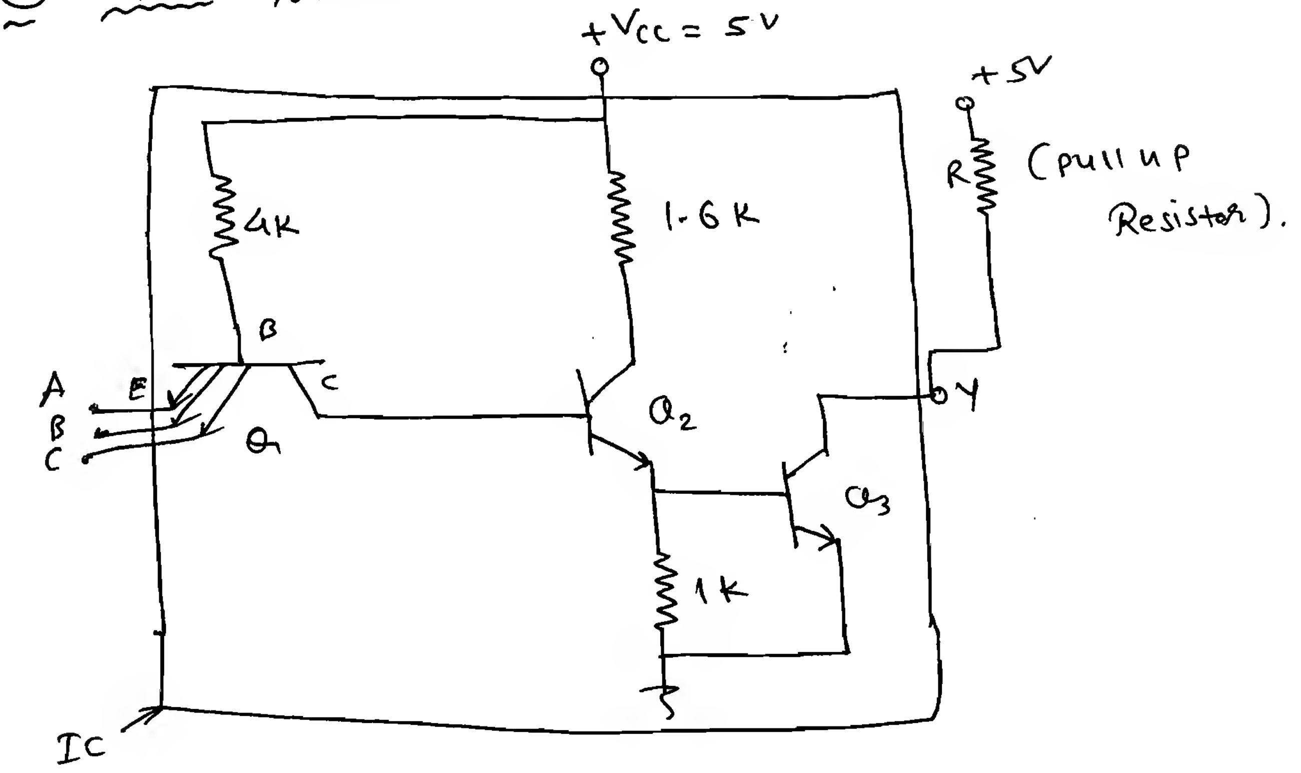
③ TTL : (Transistor Transistor Logic). (saturated).

→ ⓐ open Collector.

ⓑ Totempole.

ⓒ Tri-State.

→ ⓐ open collector.



A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

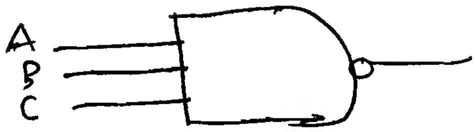
→ Q is in forward Active region mode

Q_2 ~~is~~ and Q_3 ~~are~~ are OFF ⇒ Y = +5V

→ { Q_1 is in Reverse active
Q_2 is ON , Q_3 is ON , $Y = V_{CE_3 sat}$

$\boxed{Y = 0.2V}$

→ So, it is NAND gate.

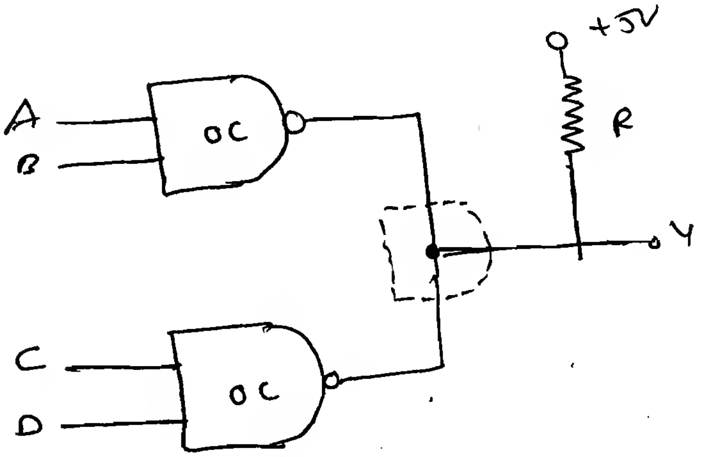


$Y = \overline{A \cdot B \cdot C}$.

* Advantuges:

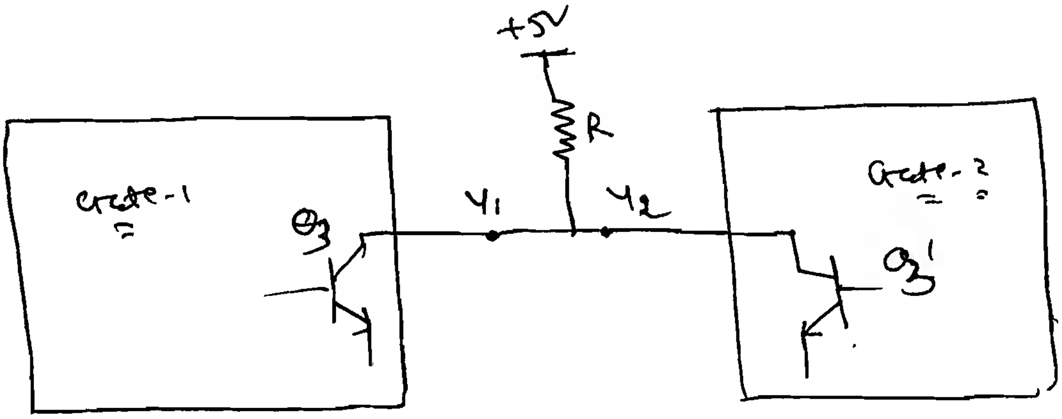
→ (1) Open Collector TTL

→ Wired - AND Logic. ✓



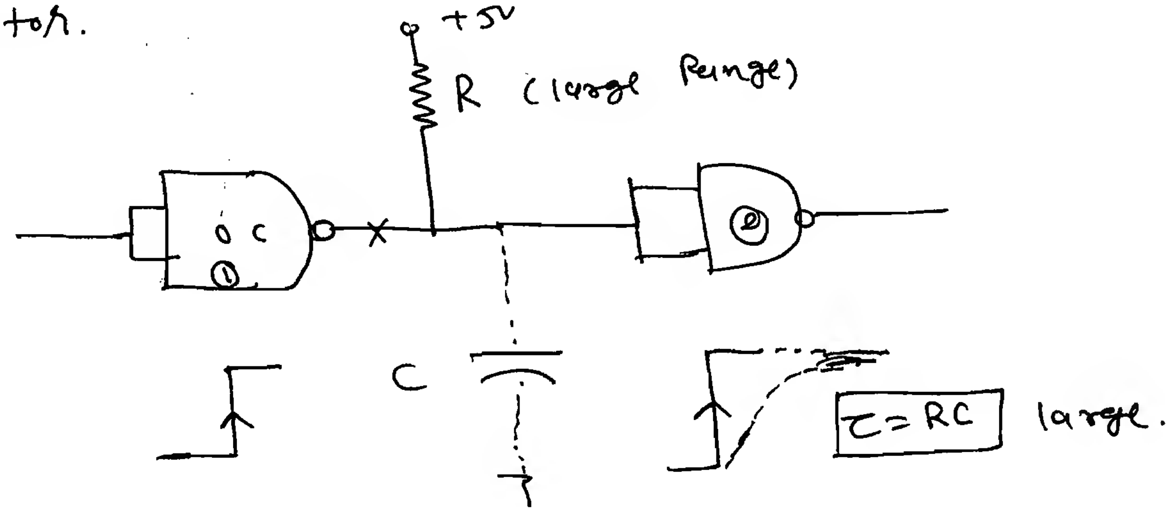
$Y = \overline{AB} \cdot \overline{CD}$

$Y = \overline{AB + CD}$.



* Disadvantage:

→ In open collector TTL the stray capacitance at the i/p of the driven gate takes more time to change the voltage levels. because of high time constant $\tau = RC$, where R = pull up Resistor.



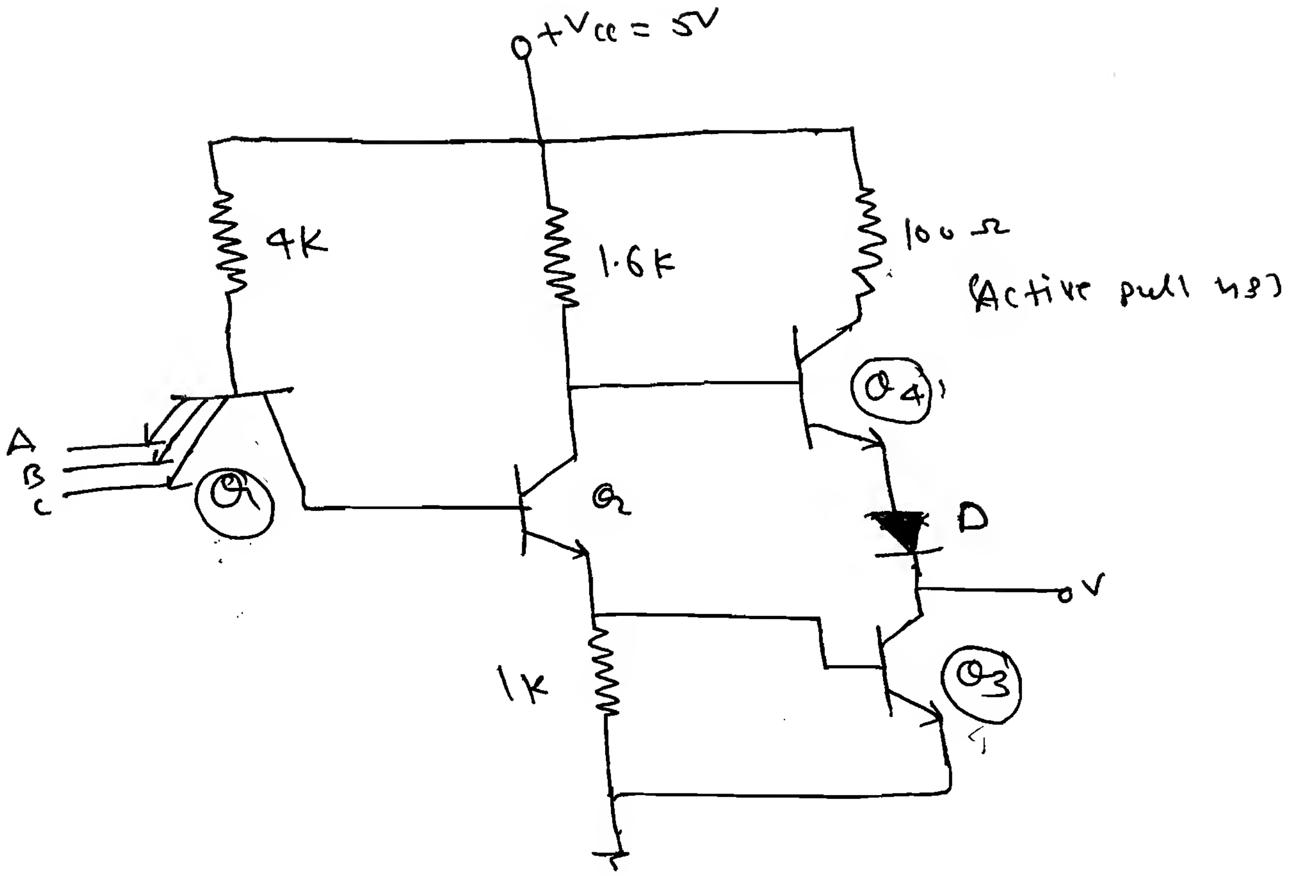
② Totempole: TTL;

→ To increase the speed of TTL the passive pull-up Resistor is Replace by active pullup which is a Emitter follower whose current gain is more and the O/P Resistance is less. Hence, the time constant is $\tau = R_o \cdot C$

~~where~~ $$\tau = R_o \cdot C$$

where, R_o = output Resistance of Emitter follower.

→



Q4 → Emitter follower

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

(0 0 0 → 1) Q1 is in Forward Active mode
Q2, Q3 are OFF ⇒ Y = +5V
Q4 ON

(1 1 1 → 0) Q1 is Reverse Active mode
Q2, Q3 are ON,
$Y = V_{CE,sat} = 0.2\,V$.
Q4 OFF, D is OFF.

→ In, TTL Logic floating inputs are considered as Logic-1.

→ Q_2 is called Phase Spiter because it will maintain the exclusive condition betn Q_3 & Q_4. that is one of them is on other one is off.

* Advantage:

→ High Speed of operation because of less time Constant $\tau = R_o \cdot C$

→ R_o = output resistance emitter follower.

* Disadvantge:

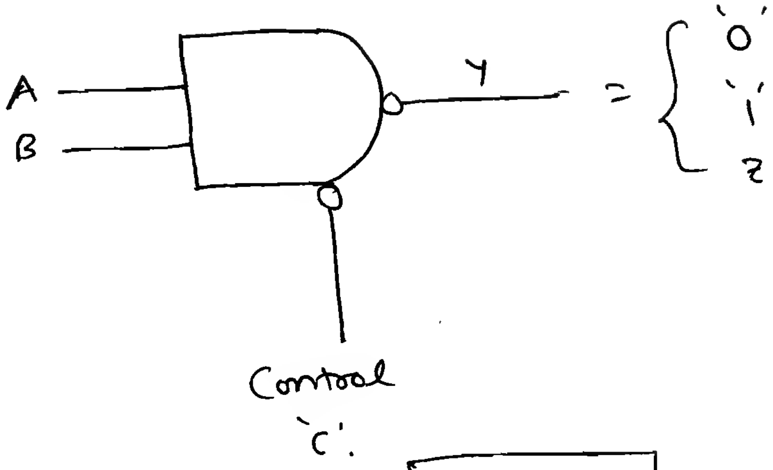
→ (i) As Switch off time is more than switch on time, for a short period of time Both Q_3 & Q_4 will be in the on Condition which results in large Current drawn from the Supply.

(ii) Wired AND logic is not possible with totempole TTL.

⇒ To improve the speed of Totem Pole TTL Q_4 and D are replaced by Darlington pair which has high current gain and very low O/P resistance.

③ Tristate TTL:

→



If $\boxed{C=0}$ => $\boxed{y = \overline{A \cdot B}}$

If $\boxed{C=1}$ => $\boxed{y = 'z'.}$ high Imp. state.

→ when $\boxed{C=1}$

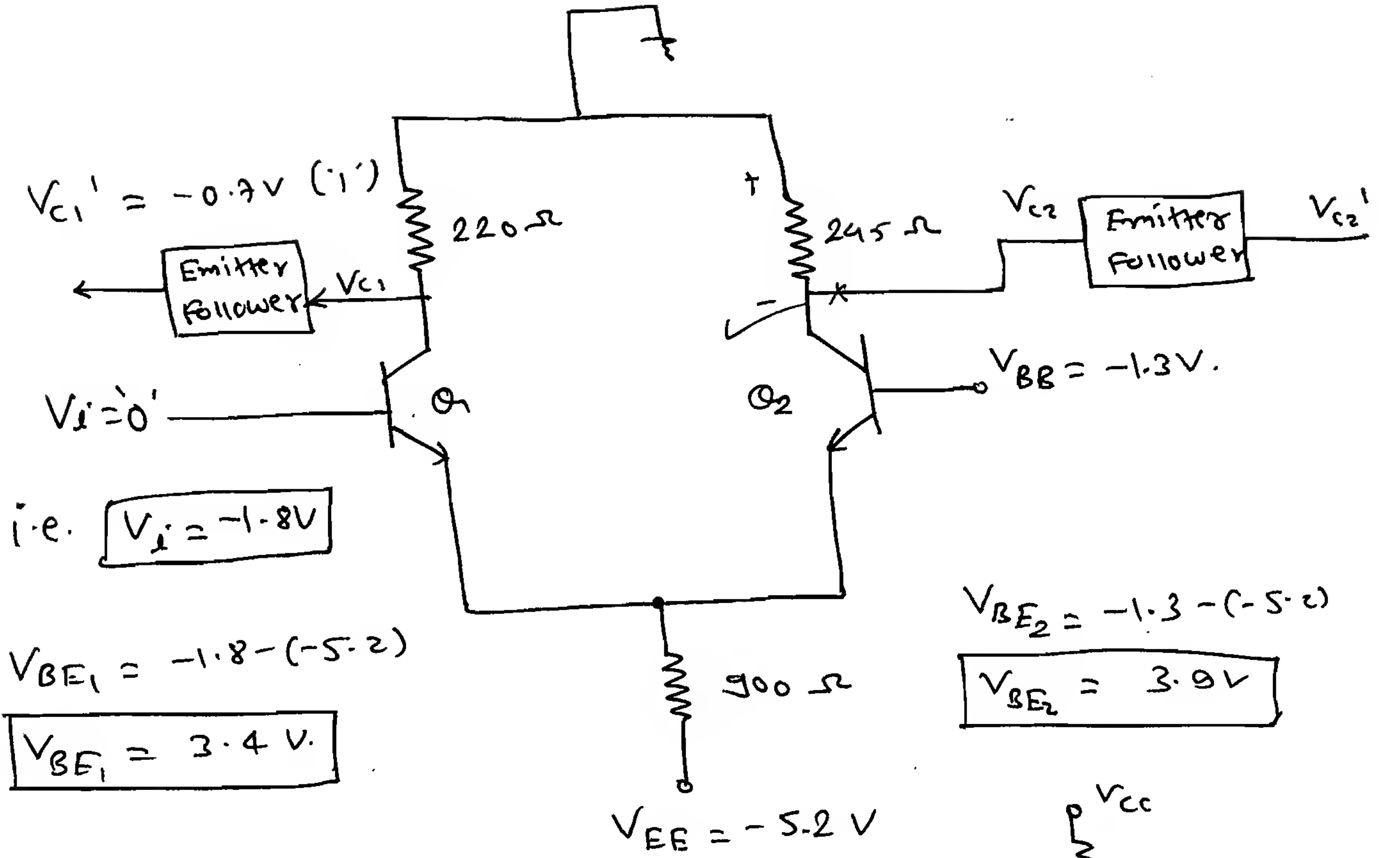
=> 'Both 'Q_3' and 'Q_4' will become OFF.

→ Tri-state TTL are used in Bus Configuration, of micro Computer.

③ ECL (Emitter Coupled Logic).

(OR) Non-Saturating Logic,

CML (Current mode logic).



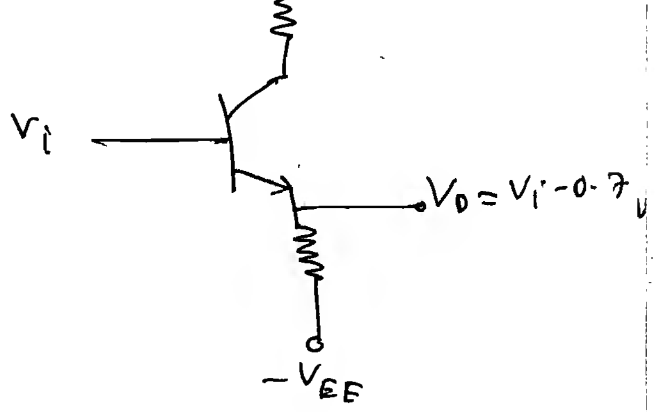
i.e. $\boxed{V_i = -1.8V}$

$V_{BE_1} = -1.8 - (-5.2)$

$\therefore \boxed{V_{BE_1} = 3.4\,V.}$

$V_{BE_2} = -1.3 - (-5.2)$

$\boxed{V_{BE_2} = 3.9V}$



$$\therefore \quad V_{C2} = -I_{C2} \cdot R_{C2}.$$

$$\therefore \quad V_{C2} \approx -I_{E2} \cdot R_{C2}.$$

$$\therefore \quad I_{E_2} = \frac{V_{BB} - V_{BE} - V_{EE}}{0.9}$$

$$\therefore \quad = \frac{-1.3 - 0.7 - (-5.2)}{0.9}$$

$$\therefore \quad I_{E_2} = \frac{3.2}{0.9} = 3.555\,mA$$

$\therefore \boxed{V_{C2} = -0.9V}$

V_i	Transistor	outputs.
'0' $= -1.8V$	Q_1 OFF Q_2 ON (Active Region)	$V_{C1} = 0$; $V_{C1}' = -0.7$ (a '1'). $V_{C2} = -0.9V$; $V_{C2}' = -0.9 - 0.7$ $= -1.6V$ ('0').

Vi	Transistor	outputs.
'1'	Q_2 OFF	$V_{c1} = -0.9V$; $V_{G}' = -1.6V$ ('0').
(−0.8V)	Q ON (Active region)	$V_{c2} = 0$; $V_{c2}' = -0.7$ ('1').

(ii) If $V_i = '1' = -0.8V$.

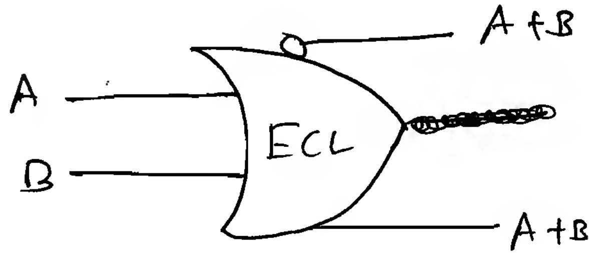
$V_{BE1} = 4.4V$;

$V_{BE2} = 3.9V$.

Hence Q conducts, Q_2 is OFF.

→ From the above table we can absorved that $\boxed{\begin{array}{l} V_{c1}' = \overline{V_i} \\ V_{c2}' = V_i \end{array}}$ &

→



*** Advantage:**

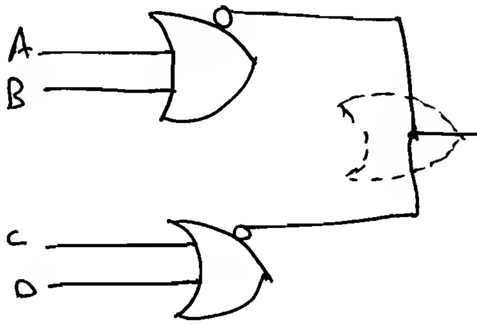
→ High Speed of operation because it is in non-Saturating logic.

→ No current spikes.

→ High Fanout (≈25).

(4) Both NOR and OR gates are available.

(5) Wired - ~~&~~ OR logic is possible.



$y = \overline{A+B} + \overline{C+D}$.

* Disadvantages:

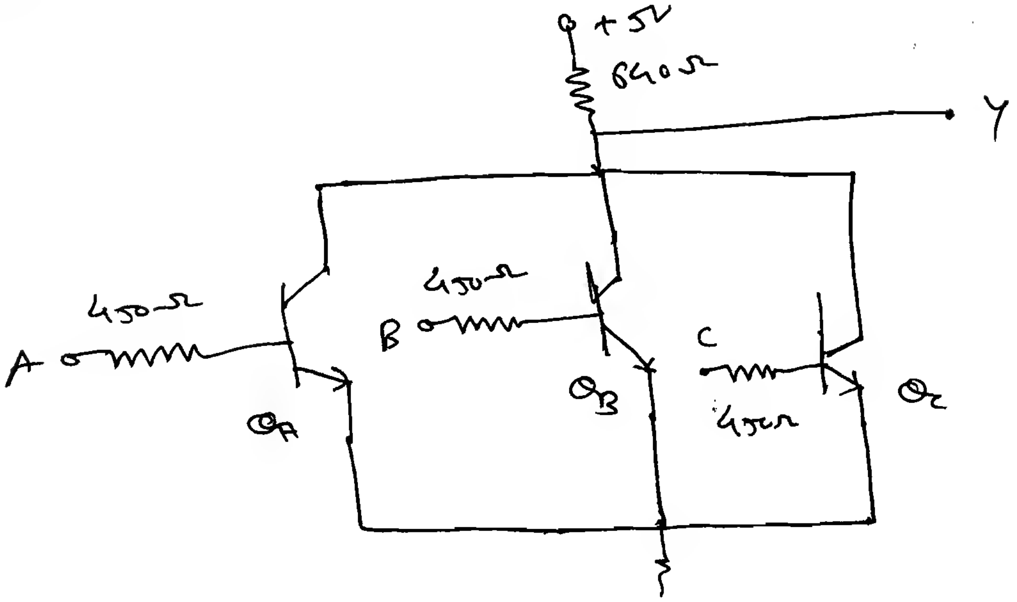
(1) High power dissipication because the Transistors operate in active region.

(2) the Noise margin of ECL gate is very less. [≈ 0.31 V] ..

Sr. No.	Parameters	Bipolar			MOSFET family	
		OTL	TTL	ECL	MOS	CMOS
1)	Fanout	8	10	25	20	>25
2)	Power Dissipation (mw)	8-12	12-22	40-55	0.2-1	0.01.
3)	Prop. Delay.	30	6-12	1-4	300	70
4)	Noise Immunity	Good	very good	Fair	Good	verygood.

→ BIMOS: Bipolar + MOSFET.

* RTL (Resistor Transistor Logic).



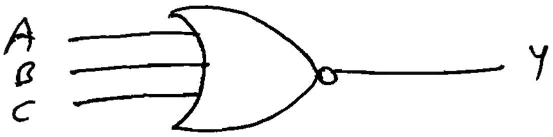
A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

→ { Q_A, Q_B, Q_C are OFF
$Y = +5V$.

→ $Y = V_{CE,sat} = 0.2V$

$Y = \overline{A+B+C}$

So, it is NOR gate.



→ RTL has more resistors so it occupies more space than MOSFET families so it is outdated.

* I²L : (Integrated Injection logic)

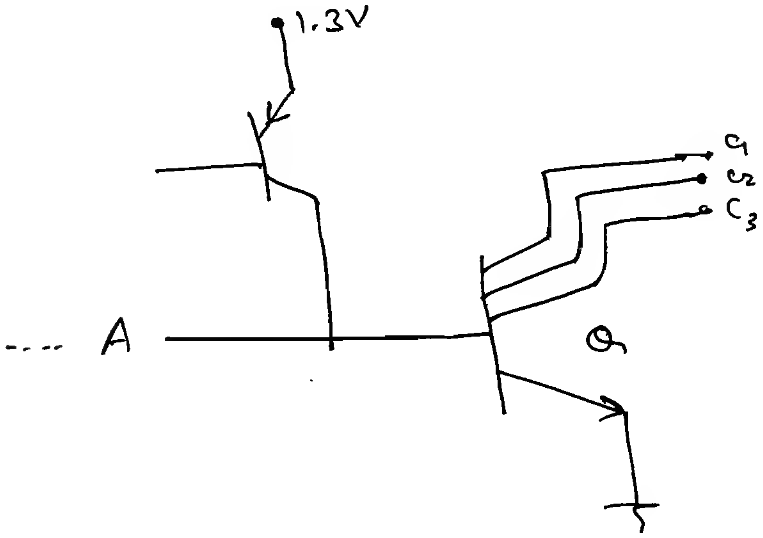
→ I²L is obtained from RTL by making three changes:

(i) Base Resistors are omitted.

(ii) PNP transistors is used in place of collector Resistor.

(iii) Multicollector transistors are used.

→



* advantages:

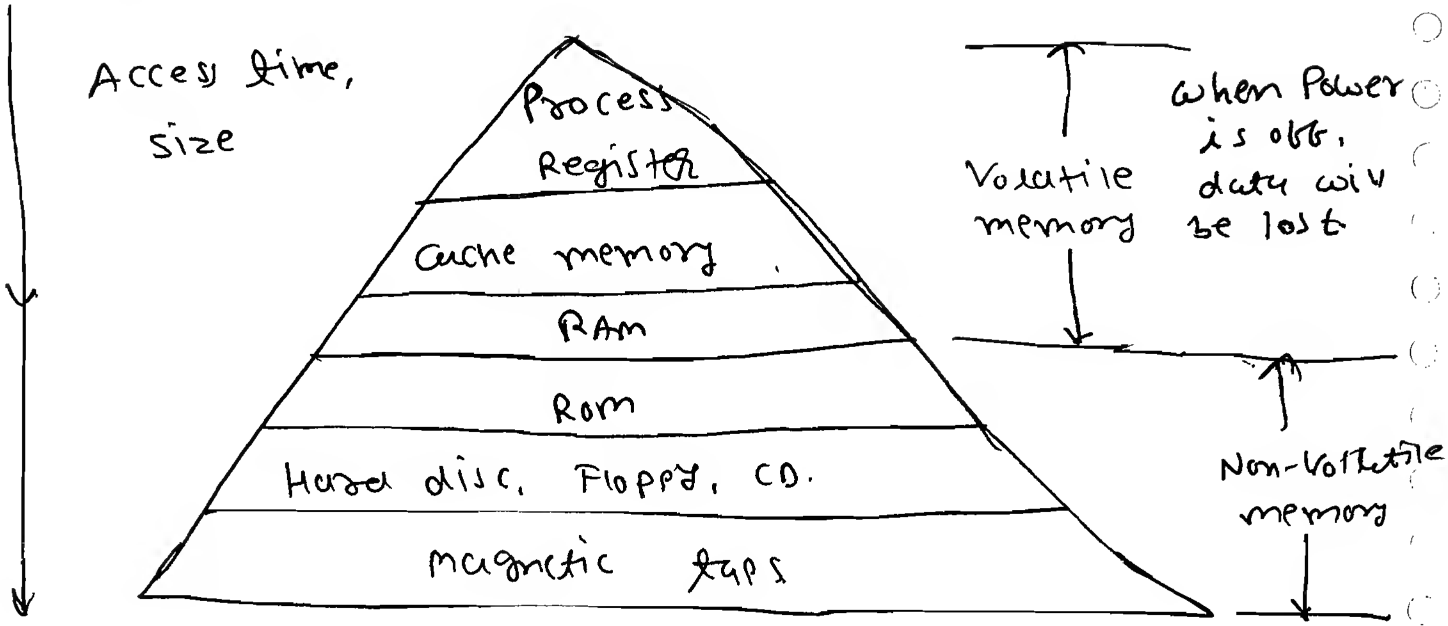
(i) High pacage density.

(ii) Low power and prop. delay product.

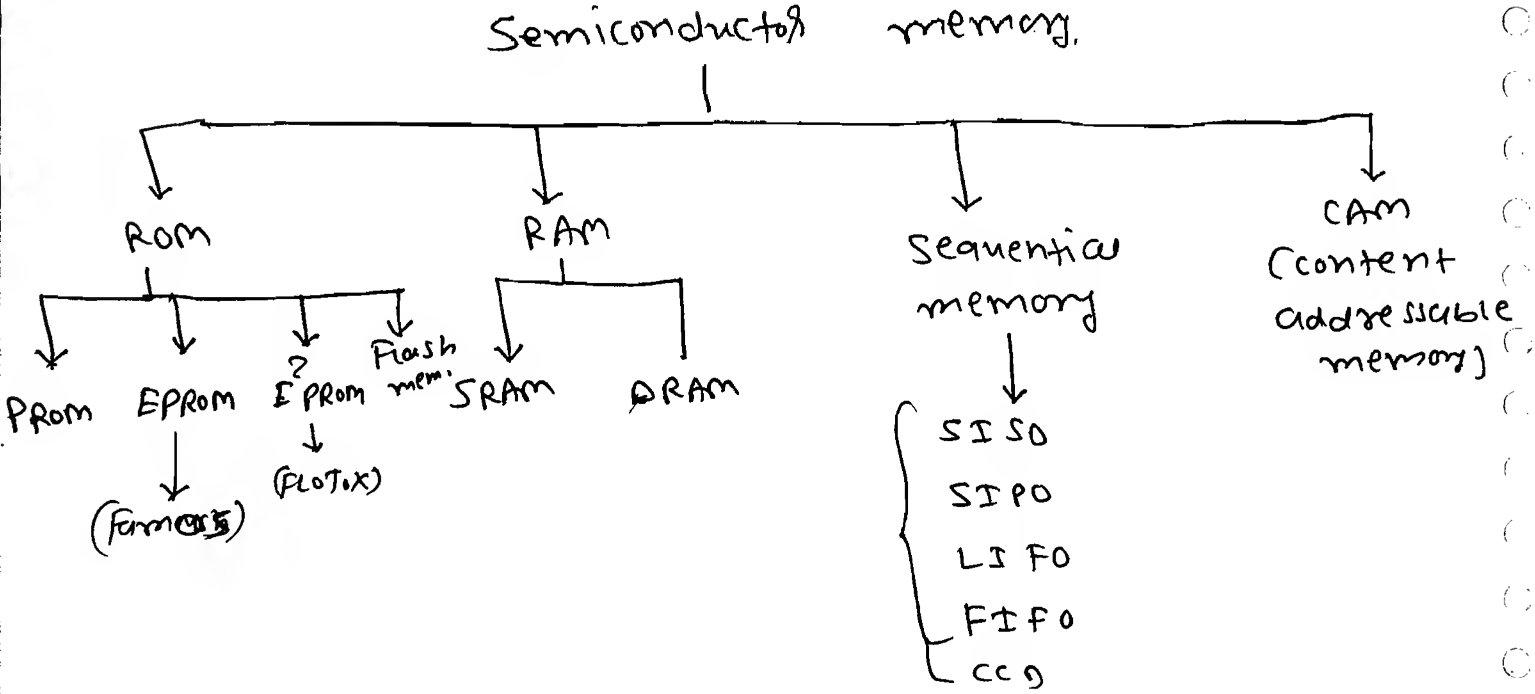
→ In I²L the speed of operation can be control by choosing the voltage input of PNP transistor appropriatly.

⇒ The High threshold logic (HTL) family is having high Noise margin (NM) ≈ 7.1V.

☆ Semiconductors memories:

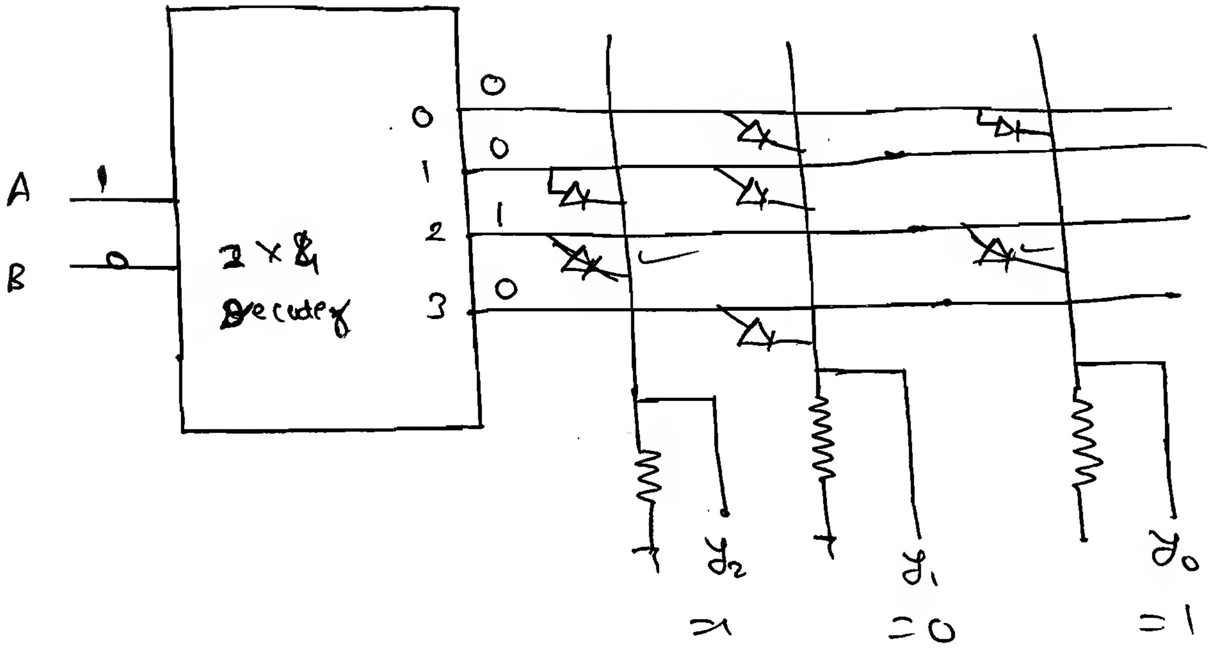


*



* Rom using Diode matrix:

→



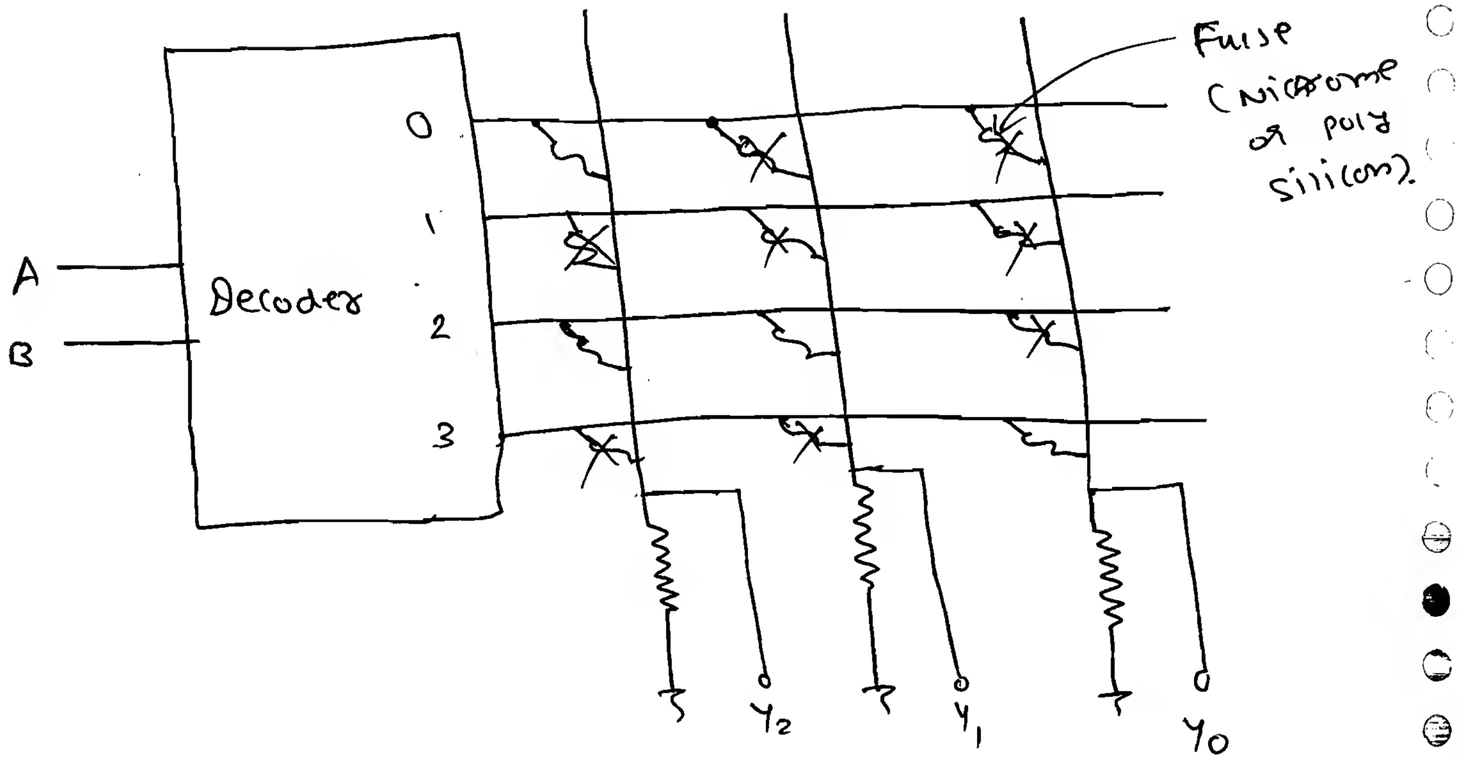
→ Rom of Combinational circuits.

$$\begin{cases} Y_2(A,B) = \Sigma m(1,2). \\ Y_1(A,B) = \Sigma m(0,1,3). \\ Y_2(A,B) = \Sigma m(0,2). \end{cases}$$

Rom size = $2^x \times y = 2^2 \times 3 = \boxed{12}$.

* PRom: (programmable Rom) (OTP Rom)

(one time programmables.

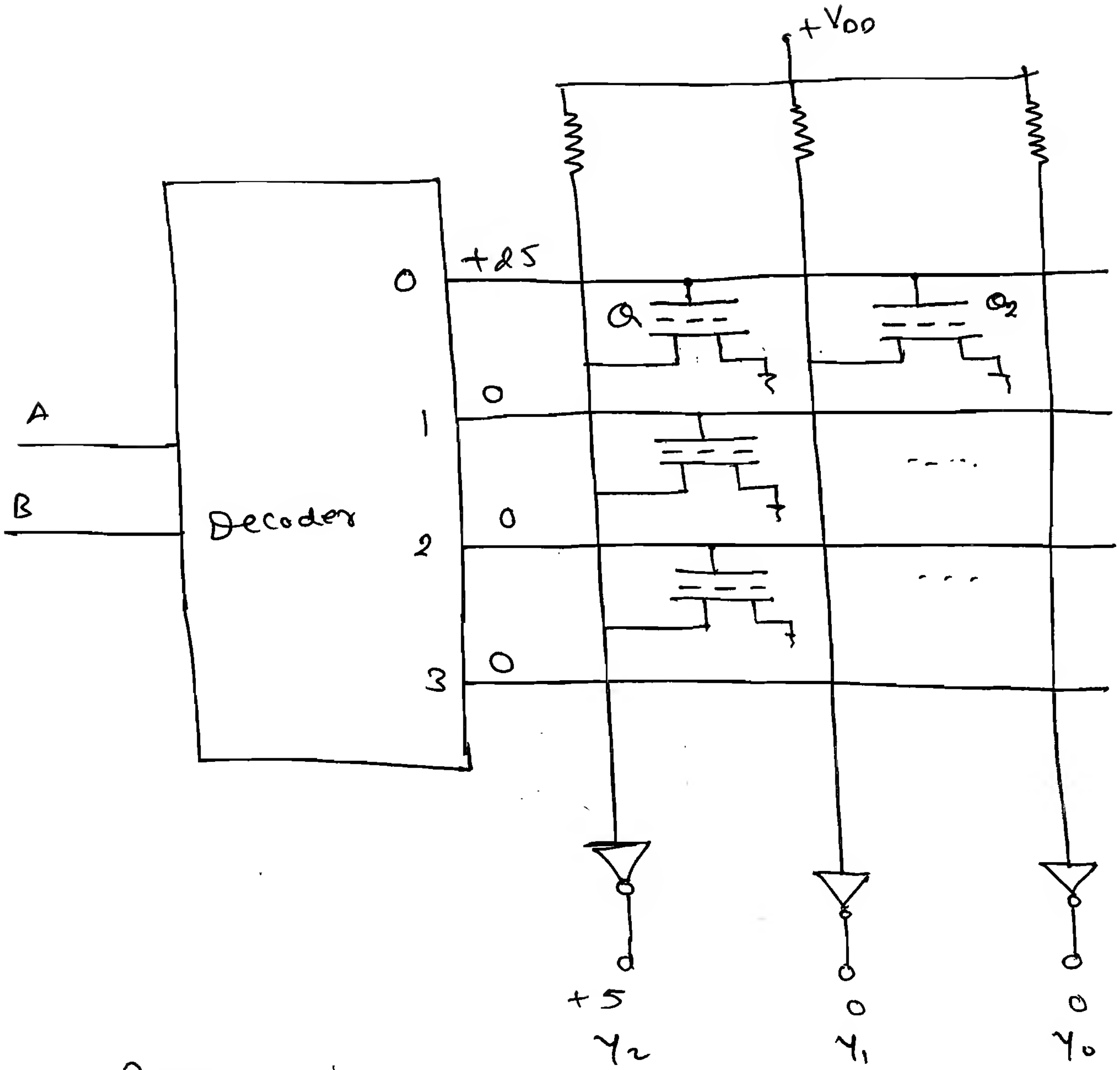


$Y_2(A,B) = \Sigma m(1,3).$

$Y_1(A,B) = \Sigma m(0,1,3).$

$Y_0(A,B) = \Sigma m(1,2,3)$

* Erasable PROM (EPROM).



(1) Programming:

Q1 → has Trapped Electrons on floating. gate
Q2 → has no trapped electrons on floating gate.

(2) Reading:

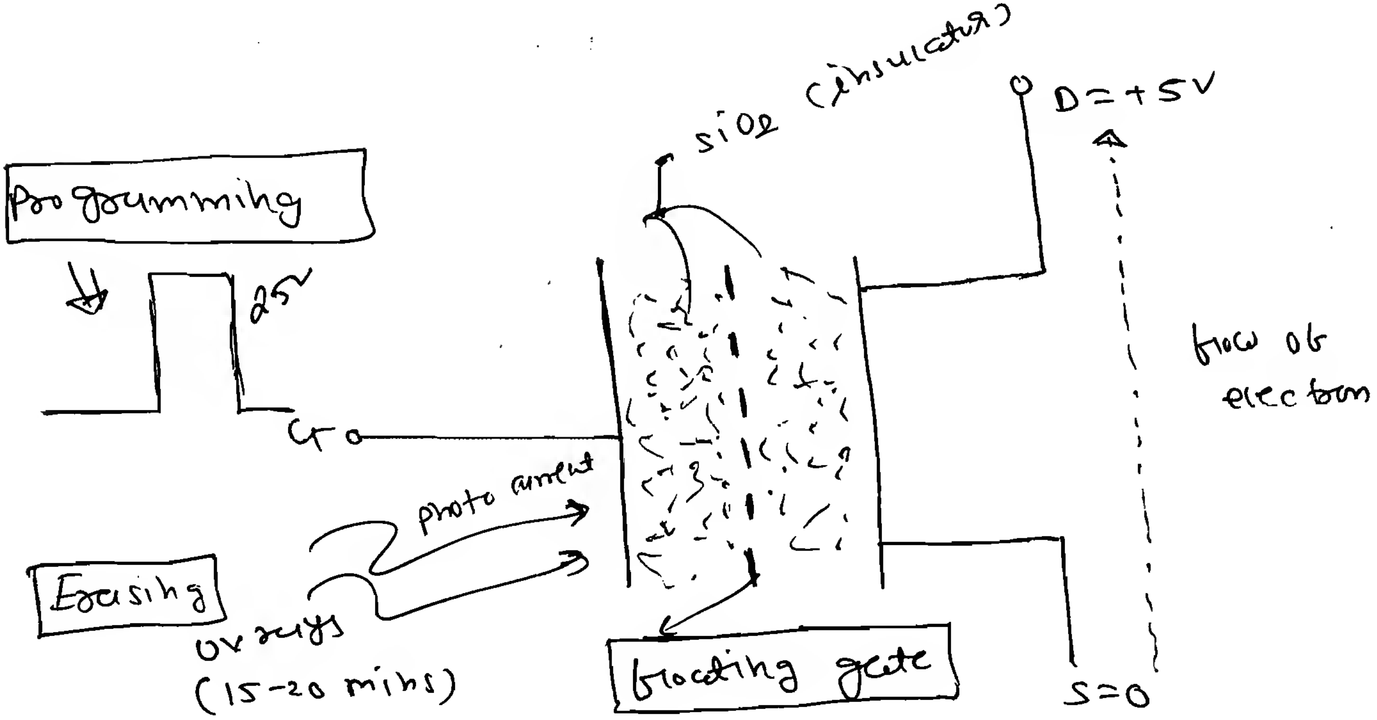
→ In a floating gate MOSFET if the electrons are trapped on the floating gate it results in Increasing in threshold voltage to 7 v.

→ Hence, For 5 v Q1 is going to be off and Q2 is on and corresponding outputs are

$Y_2=0$ and $y_1=1$.

③ Erasing:

→ To earase we have to use ~~Ph~~ UV rays bombard into it so that the electrons goes into their normal condition.



* Disadvantages:

(i) Insystem programming (ISP) not possible.

(ii) Erasing takes more time.

(iii) Partical erasing is not possible.

⇒ other name of it is

FAMOS

⇓

Floating gate Avalanche injected MOS FET.

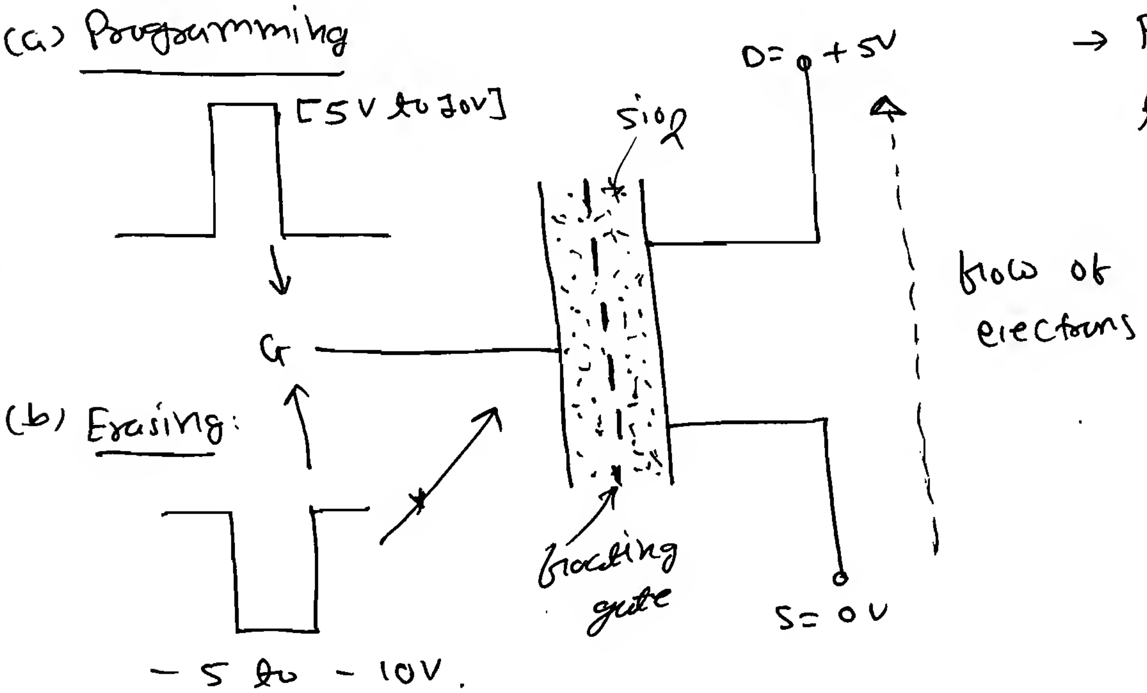
* E^{2}PROM (Electricaly Erasable PROM).

→ In E^{2}PROM the thickness of silicon oxide layer in floating gate MOSFET is Reduced because of this, 5 to 10 V of Programming pulse is sufficient to scape the electrons on the floating gate.

→ The MOSFET can be erased by reversing the polarity at the gate input.

FLOTAX

→ Floating gate tunneling oxide MOSFET.



* Advantages:

(1) Partial erasing is possible.

(2) ISP (In system programming) is possible.

(3) Erasing takes very less time.

* Disadvantages:

→ It has low package density because each bit requires 2 MOSFETS one floating gate MOSFET and another one is ordihary MOSFET.

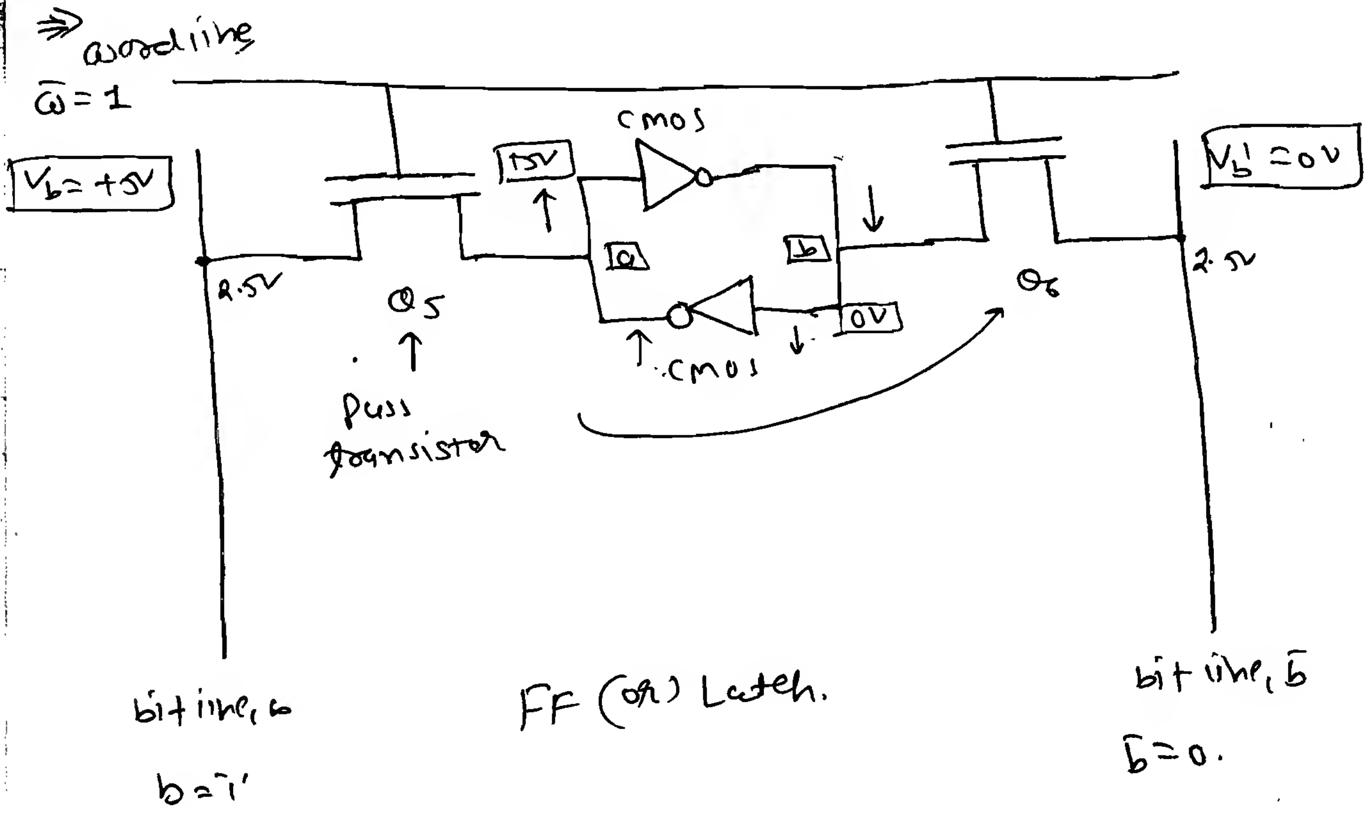
(2) High cost per bit.

* Flash Memory:

→ Advantages of Flash memory.

1. High package density
2. Partica Erasing
3. ISP is possible.
4. Low cost.
5. More no. of Read/write cyles.

(a) SRAM (1bit mem. cell). (Static RAM)



→ To store Logic '1' in SRAM cell.

1. bit lines are precharged to $\boxed{2.5V}$
2. Choose $\boxed{W=1}$; $\boxed{V_b = +5V}$; $\boxed{V_b' = 0V.}$
3. 'V_a' rises to +5V ; V_b drops to 0V.

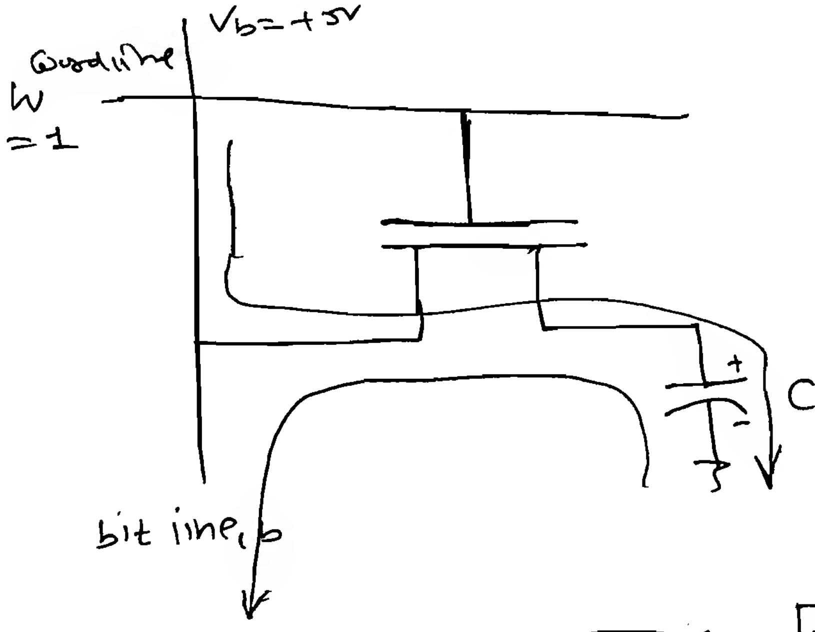
Thus logic '1' is store in SRAM cell.

→ To Read SRAM cell.

1. Choose W=1.
2. Q_5, Q_6 MOSFET are ON.
3. Hence, b='1' , $\bar{b}$=0.

(2) DRAM: (Dynamic RAM).

⇒



Refresh circuit

a) To write: Choose $\boxed{W=1}$; $\boxed{V_b = +5V}$ ⇒ Logic '1' is stored.

b) To Read : Choose $\boxed{W=1}$

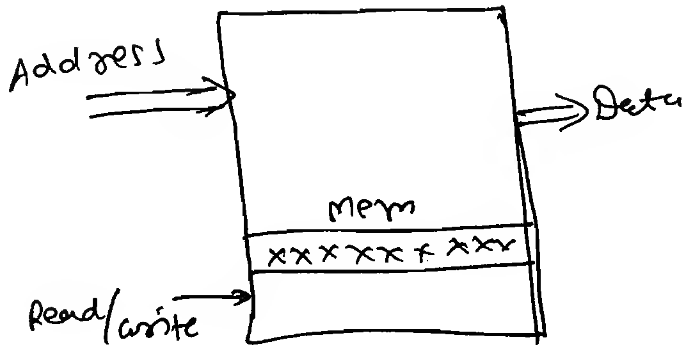
⇒ If bit line = +5V ⇒ Logic '1'.

⇒ If bit line = 0V ⇒ Logic '0'.

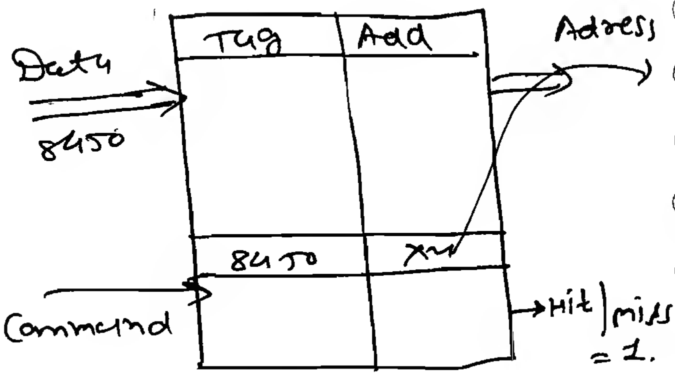
	SRAM	DRAM
Advantges:	① Speed is very high. (cache memory). ② Less hardware. Complecity.	① High package density. ② Low power Consumption. ③ Less cost
Disadvantges:	① Low package density ② High power Consumption ③ High cost	① Speed is less. (primary memory) ② more Hardware complexity. (because the voltage across Capacitor has to be refreshed at regular interval of time 2-4ms by using Refreshing ckt.

☆ CAM :

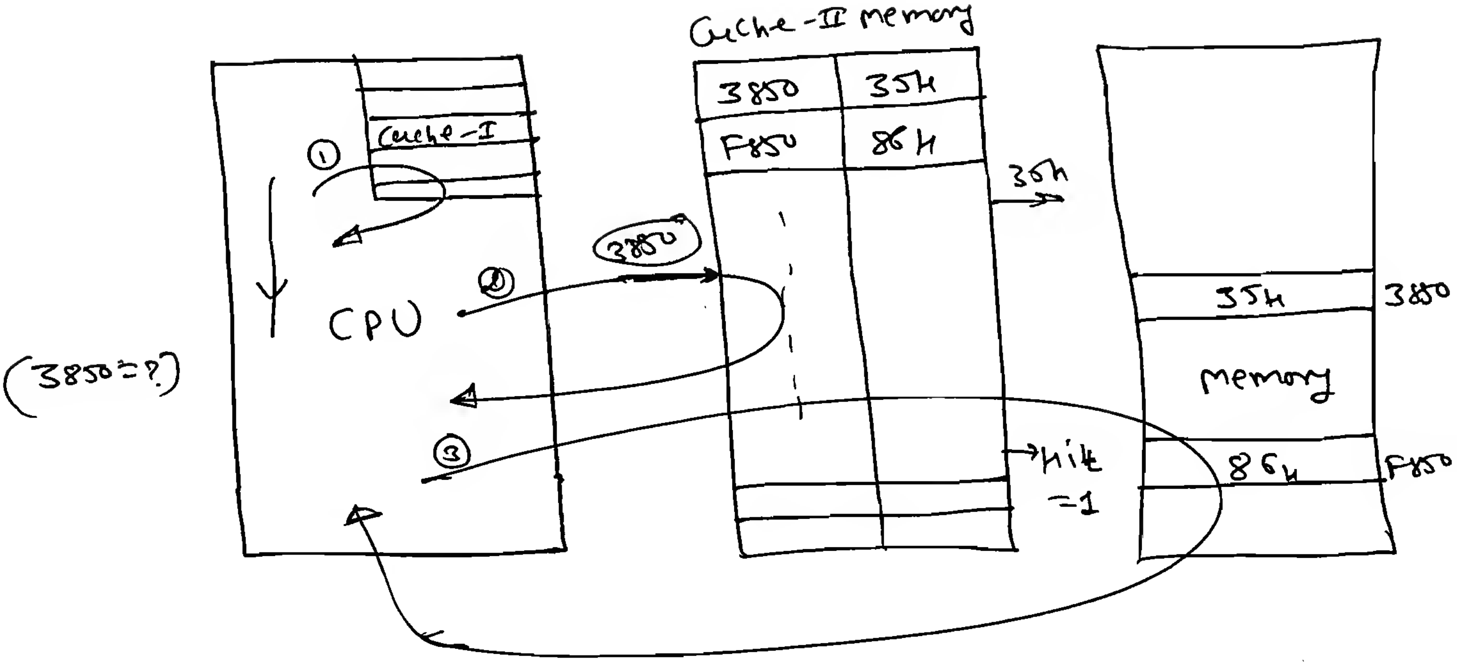
(a)



(b)



⇒ Cache memory:



☆ Digital to Analog Converter (DAC).

(a) Binary weighted Resistor DAC.

(b) R-2R Ladder DAC.

☆ Analog to Digital Converter (~~DAC~~ ADC)

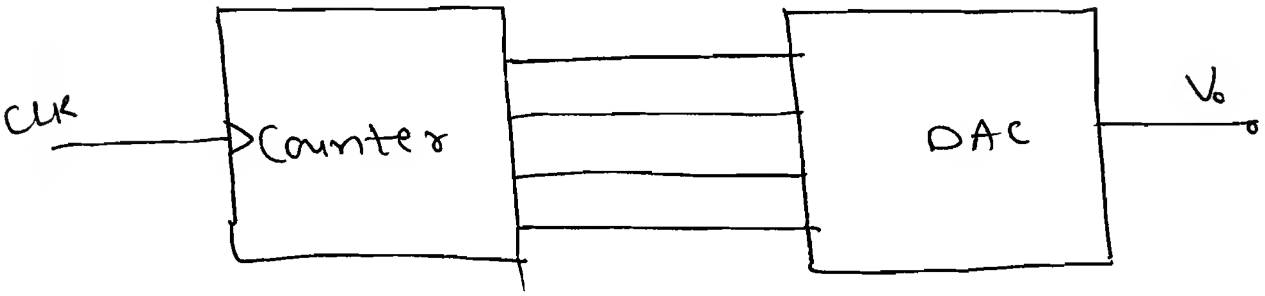
(a) Counter Type ADC

(b) Successive Approx ADC.

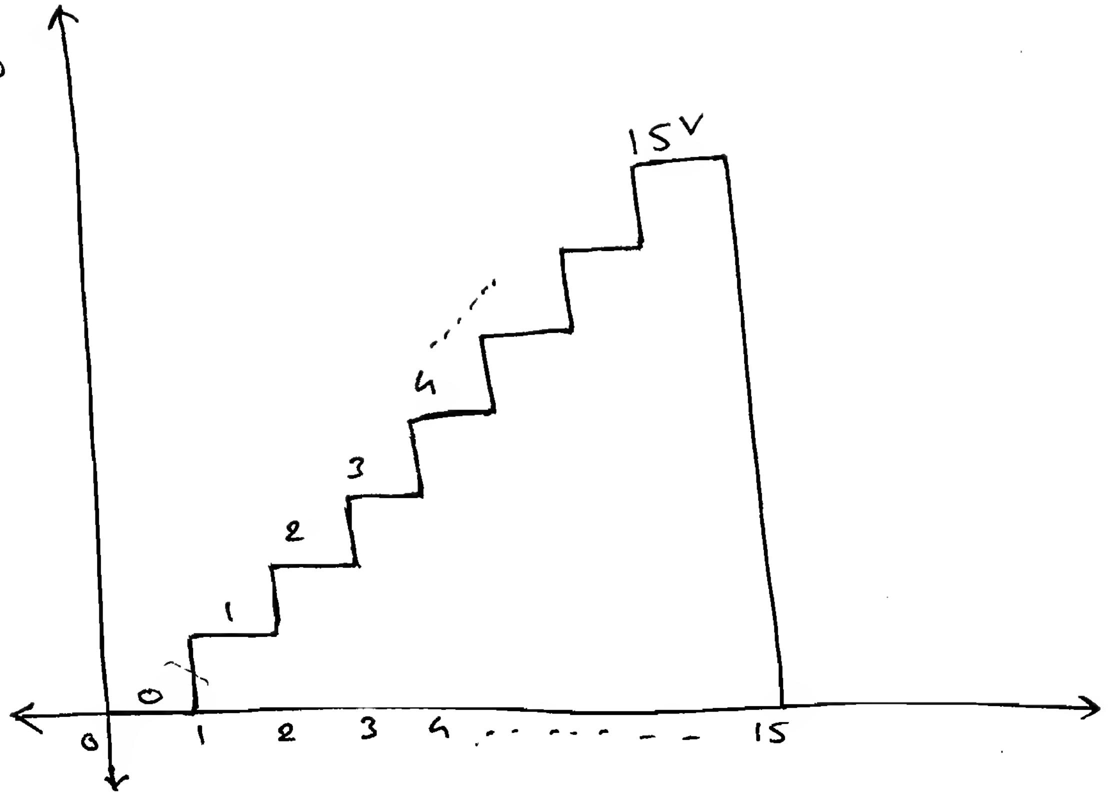
(c) Flash (OR) Parallel ADC.

(d) Dual Slope (OR) Integrating ADC.

(e) signal - delta ADC (Σ-Δ ADC).



CLK	Counter	V_0
0	0000	0V
1	0001	(let) 1V
2	0010	2V
3		⋮
⋮		⋮
15	1111	15V
16	0000	0V



(1) Fso (Full Scale O/P) = 15V

→ <u>N-Bit</u> <u>DAC</u>: = $(2^N - 1) \times$ stepsize.

(2) (a) Resolution:

→ It is the minimum Change possible at the O/P of the DAC for any Change in the digital input.

ⓐ Resolution = stepsize (Volts).

ⓑ % Resolution = $\frac{\text{Stepsize}}{\text{Fso}} \times 100$

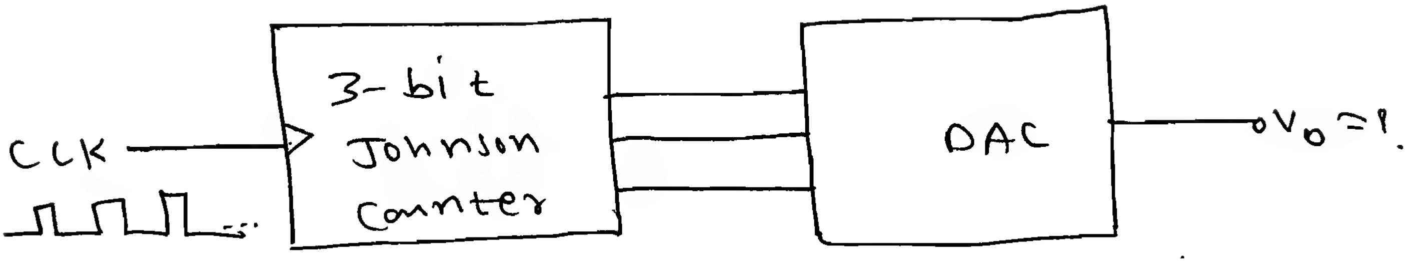
$$= \frac{\text{stepsize}}{(2^N - 1) \times \text{stepsize}} \times 100$$

$$= \frac{100}{2^N - 1}.$$

; where N = size of DAC.

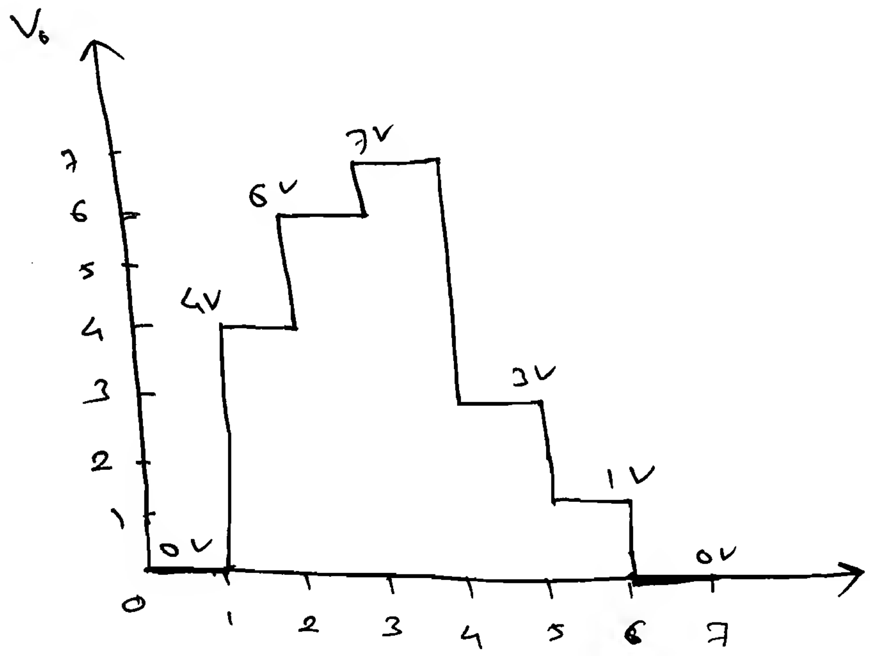
DAC	DAC
✓0.1 V	8 - Bit
0.5 V	16 - Bit
1.0 V	32 - Bit. ✓

Ex 1

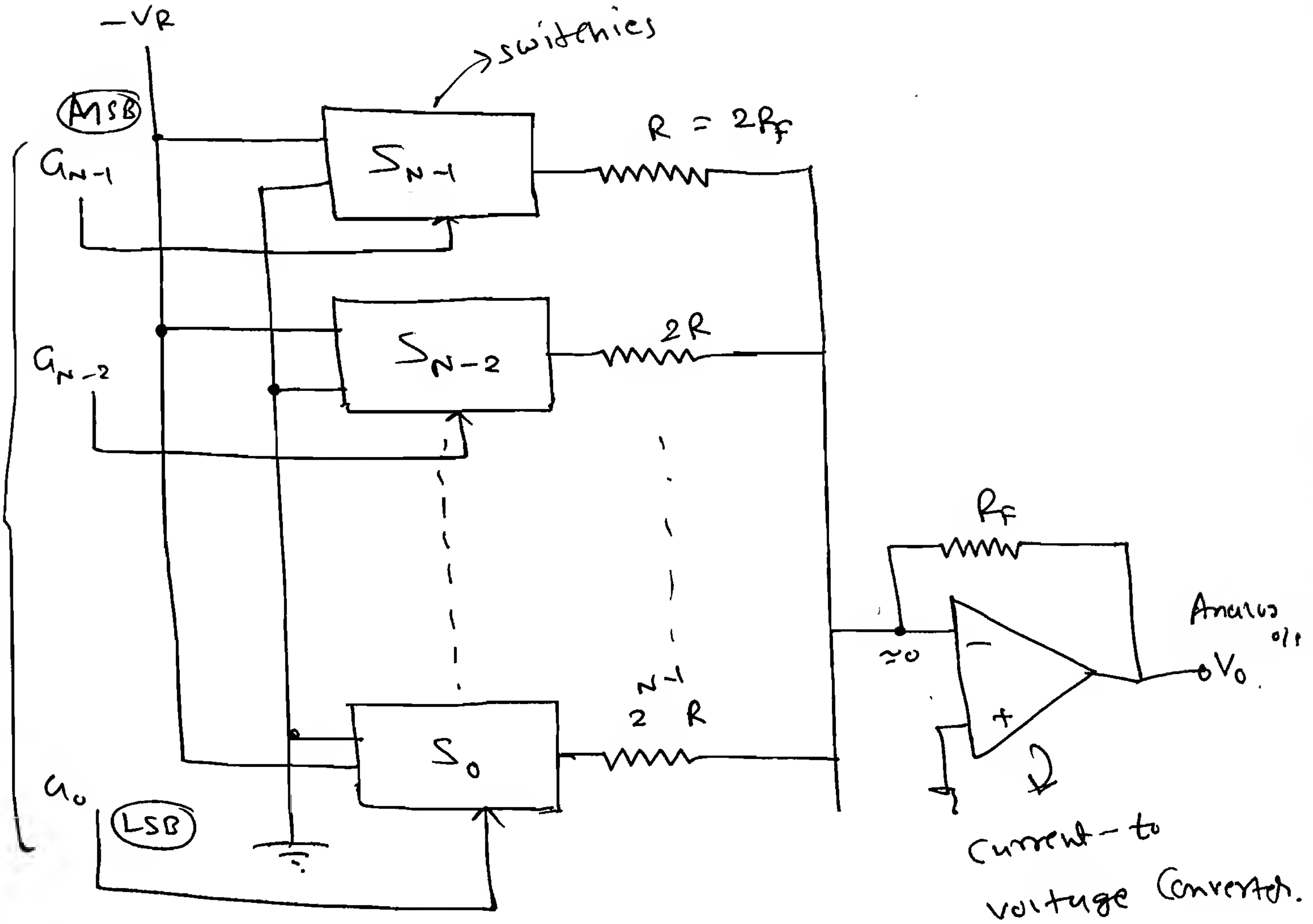


CLK | Counter | V₀.

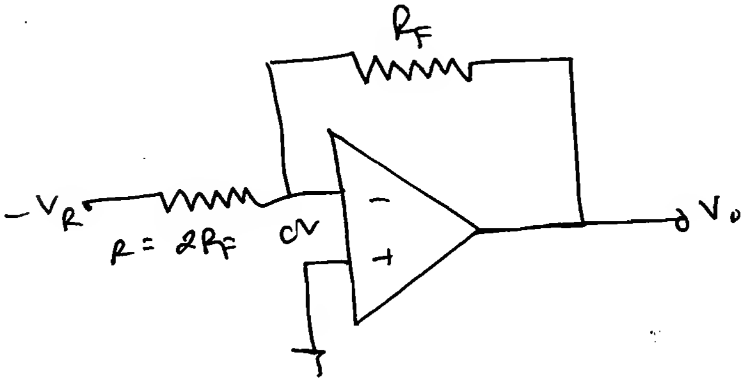
CLK	Counter	V (V)
0	0 0 0	~~1000~~ 0
1	1 0 0	4
2	1 1 0	6
3	1 1 1	7
4	0 1 1	3
5	0 0 1	1
6	0 0 0	0



(a) Binary Weighted Resistor DAC:



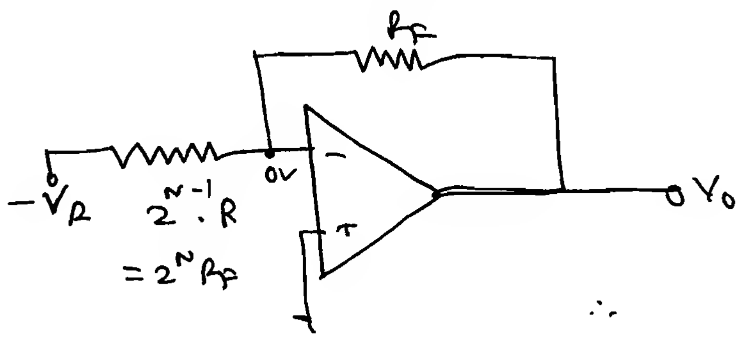
① Let, $a_{N-1} = 1$; other Input bits = 0 ⇒ $V_o = ?$



$$\therefore \quad V_o = (-V_R)\left(-\frac{R_F}{2R_F}\right)$$

$$\therefore \boxed{V_o = +\frac{V_R}{2}}$$

② Let $a_0 = 1$; other Input bits = 0 ⇒ $V_0 = ?$



$$V_0 = (-V_R)\left(-\frac{\cancel{R_f}}{2^N \cdot \cancel{R_f}}\right)$$

$$\therefore \boxed{V_0 = \frac{V_R}{2^N}} \text{ Volts.}$$

Resolution = stepsize.

Disadvantages:

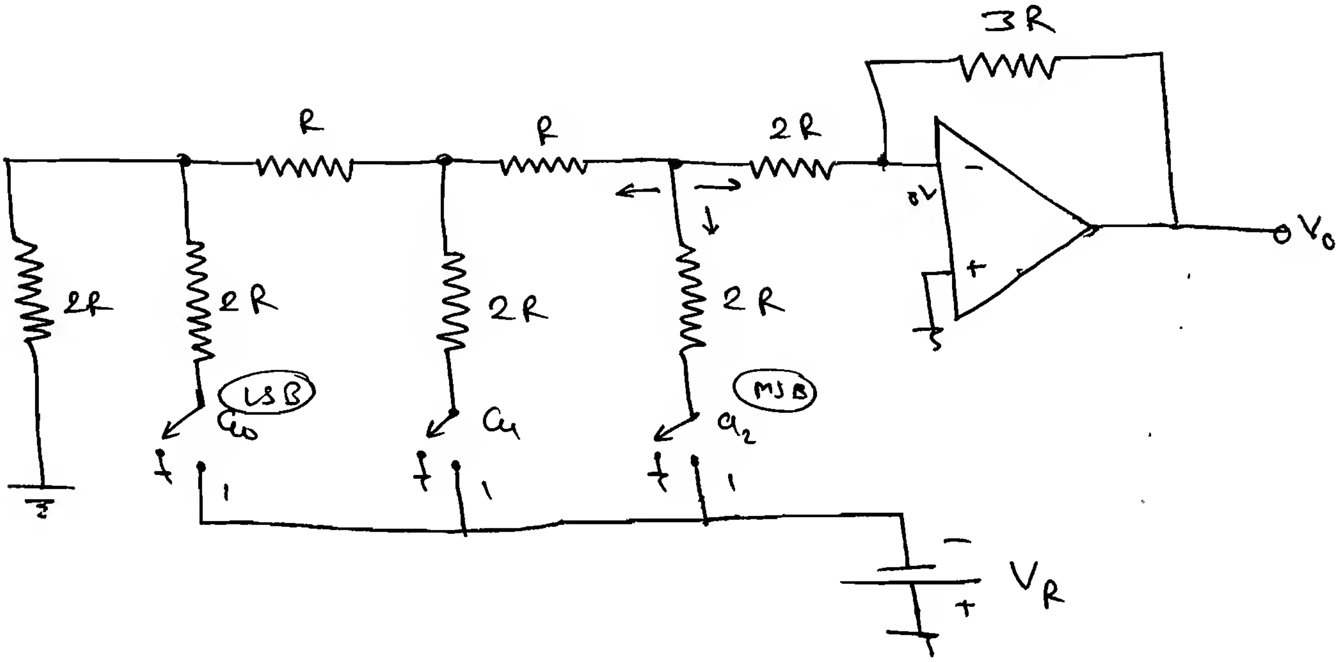
① It requires wide range of resistors.

② Using this DAC the stepsize may not be constant because it is difficult to maintain the ratio of two consequtive resistor equal to constant practically.

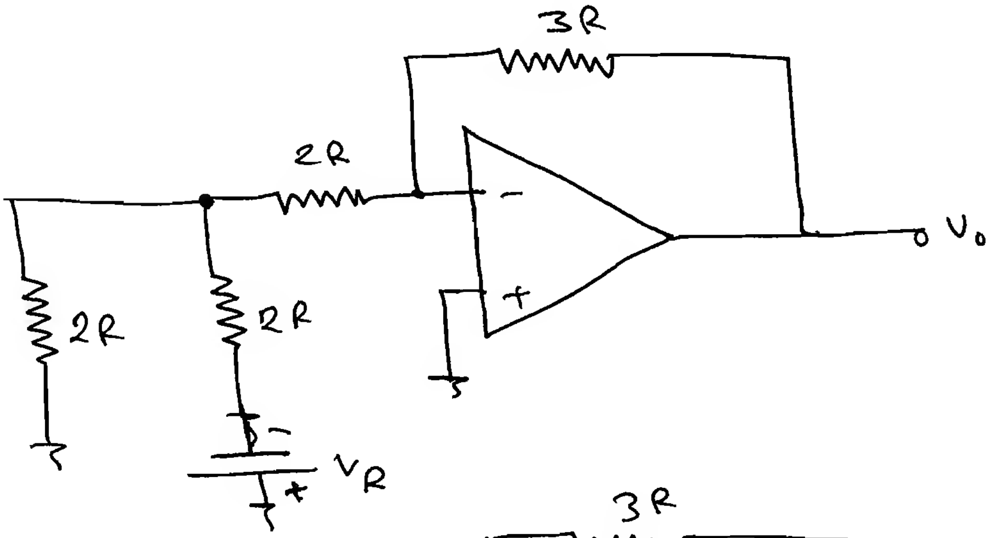
In general,

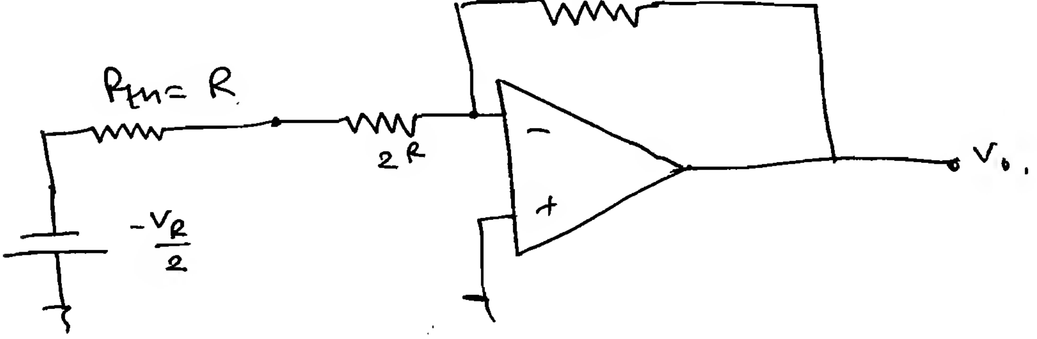
$$\therefore \boxed{V_0 = V_R\left[\frac{a_0}{2^N} + \frac{a_1}{2^{N-1}} + \cdots + \frac{a_{N-1}}{2}\right].}$$

② 3-Bit R-2R Ladder DAC (Voltage Switched)



① Let $a_2 = 1$; $a_1 = a_0 = 0 \Rightarrow V_0 = ?$

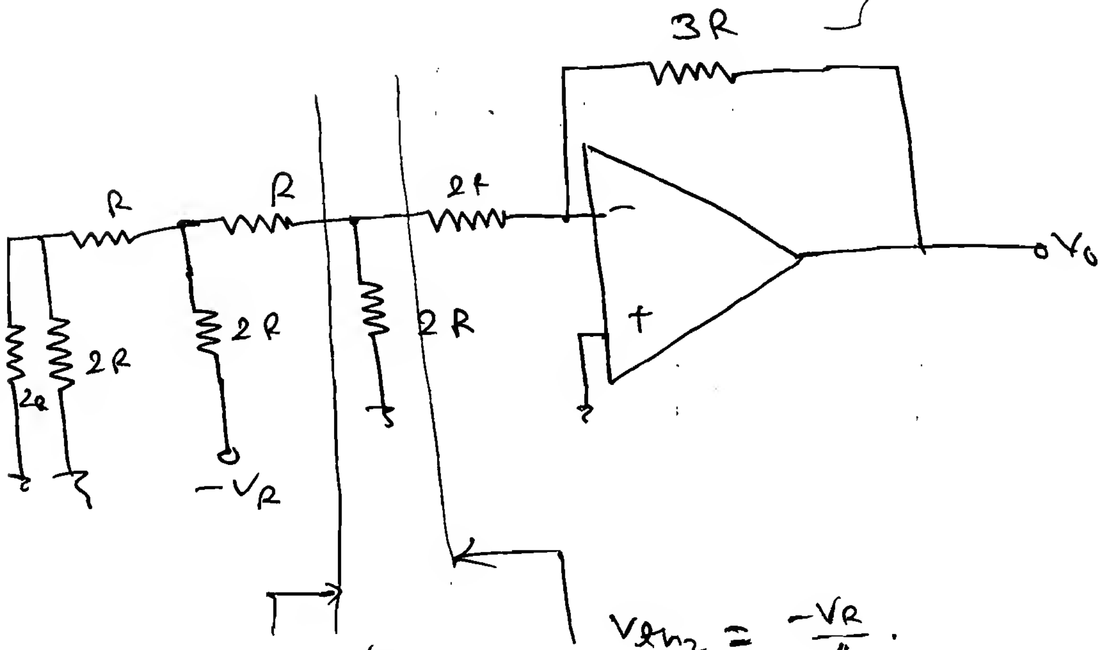




$$\therefore V_0 = \left(-\frac{V_R}{2}\right)\left(-\frac{3R}{3R}\right).$$

$$\therefore \boxed{V_0 = \frac{V_R}{2}}$$

② Let $a_1 = 1$; $a_2 = a_0 = 0 \rightarrow V_0 = ?$

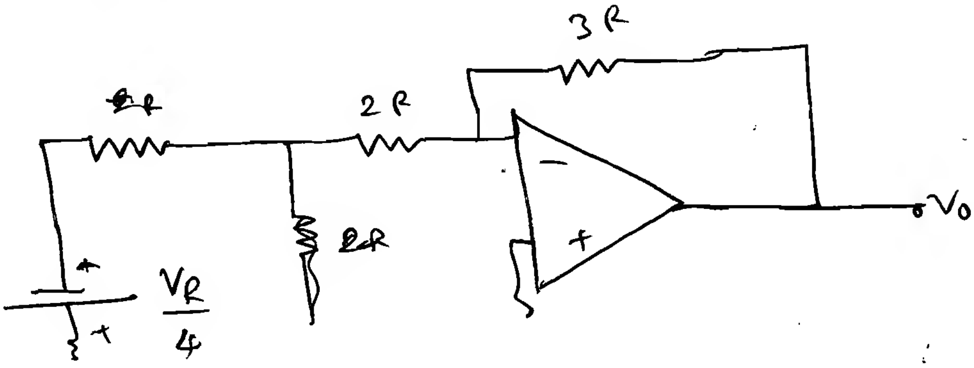


$V_{th_1} = -\frac{V_R}{2}$.

$R_{th_2} = 4R$.

$V_{th_2} = \frac{-V_R}{4}$.

$R_{th_2} = 2R$.



$\therefore \quad V_0 = \left(-\frac{V_R}{4}\right)\left(-\frac{3R}{3R}\right).$

$\therefore \quad \boxed{V_0 = \frac{V_R}{4}}$

$\therefore \quad \boxed{V_0 = \left[\frac{\overset{MSB}{a_2}}{2} + \frac{a_1}{4} + \frac{\overset{LSB}{a_0}}{8}\right] V_R}$

① $\boxed{FSO}$

$V_o = FSO$ when $a_2 = a_1 = a_0 = 1$

i.e. $FSO = \frac{V_R}{2} + \frac{V_R}{4} + \frac{V_R}{8} = \frac{7V_R}{8}$

$= V_R - \frac{V_R}{8}$.

~~③~~

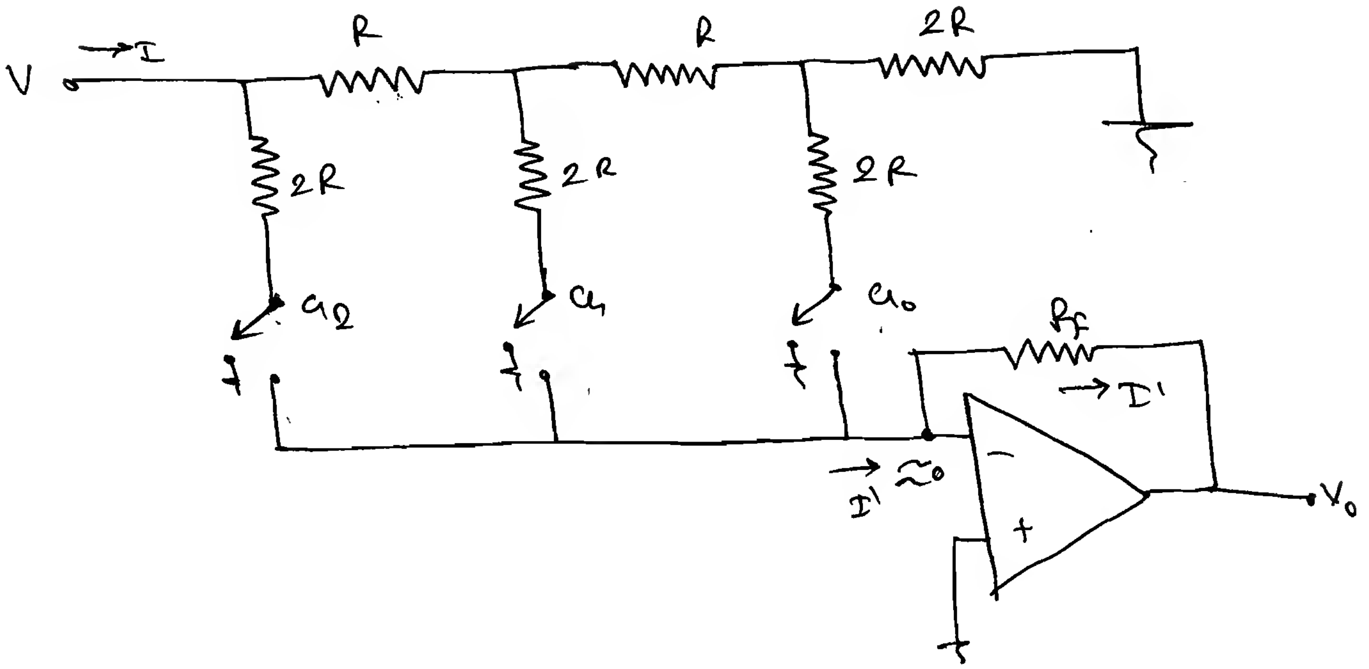
$\boxed{FSO = V_R - \text{weight of LSB.}}$

∴ ② stepsize = Resolution $= \frac{a_0}{8} V_R = \frac{V_R}{8}$.

* Disadvantage of Voltage switched Ladder DAC is: it produces unwanted spikes at the input while switching. To overcome this we use current switched DAC.

③ 3-Bit R-2R Ladder DAC (Current Switched).

=>



→ Duty of Op-Amp:

→ It is acting as Current to Voltage Amp.

→ $I = \frac{V}{R_{eq}} = \frac{V}{R} \longrightarrow ①$

$V_o = -I' R_F \longrightarrow ②$

P) Let $V = 10V$. ; $R = 1k\Omega$. $2R = R_F$

Dig. i/p $a_2 a_1 a_0 = 101$ => Analog o/p = ?

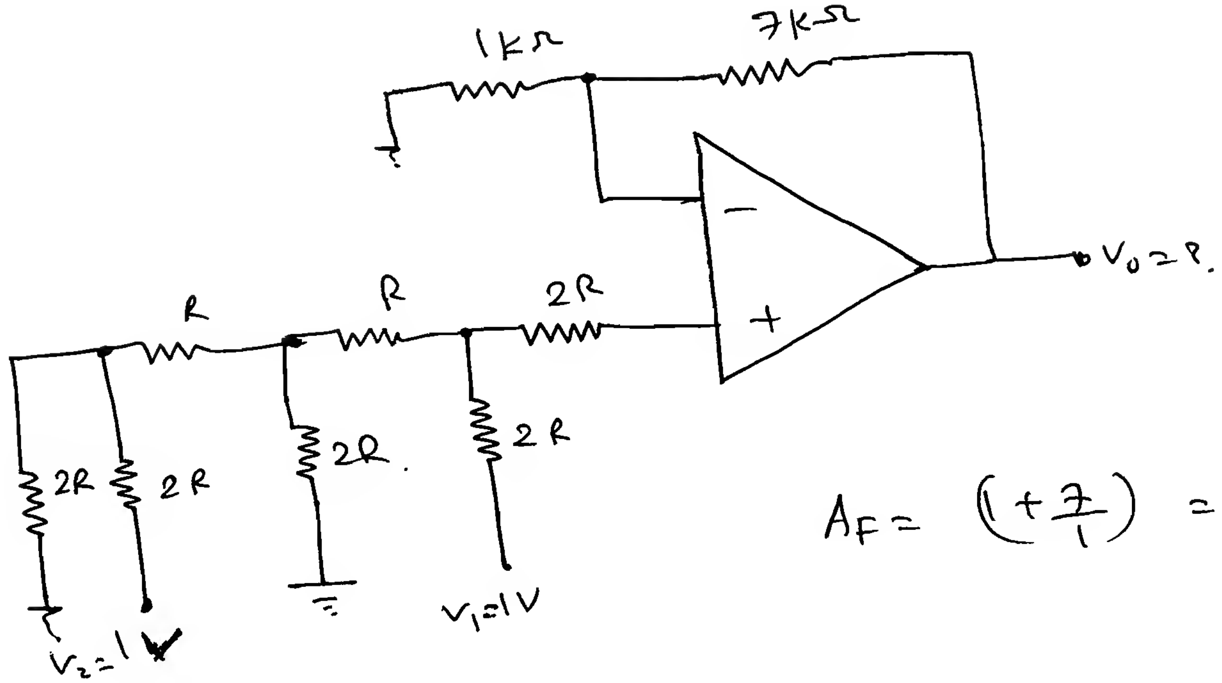
$\therefore I = \frac{V}{R} = 10mA$.

$\therefore I' = \frac{I}{2} + \frac{I}{8} = 10\left[\frac{1}{2} + \frac{1}{8}\right] = \frac{5 \times \cancel{10}\,5}{\cancel{8}\,4} = \frac{25}{4}$.

$\therefore V_o = -\frac{25}{4} \cdot (1)$.

$\boxed{V_o = \frac{-25}{4}}$ V. $\boxed{V_o = -6.25V}$

P-2) Determine the O/P of the following DAC.45



$A_F = \left(1 + \frac{7}{1}\right) = 8.$

$\therefore \quad V_o = A_F\left(\frac{1}{8} + \frac{1}{2}\right).$

$V_o = 8 \times \frac{5}{8}$

$\therefore \boxed{V_o = 5V}$

$\therefore$ Contribution of V_1 at Input $= \frac{V_R}{2} = \frac{1}{2}.$

" " V_2 at Input $= \frac{V_R}{8} = 1/8.$

$\therefore$ output $=$ Input x Gain

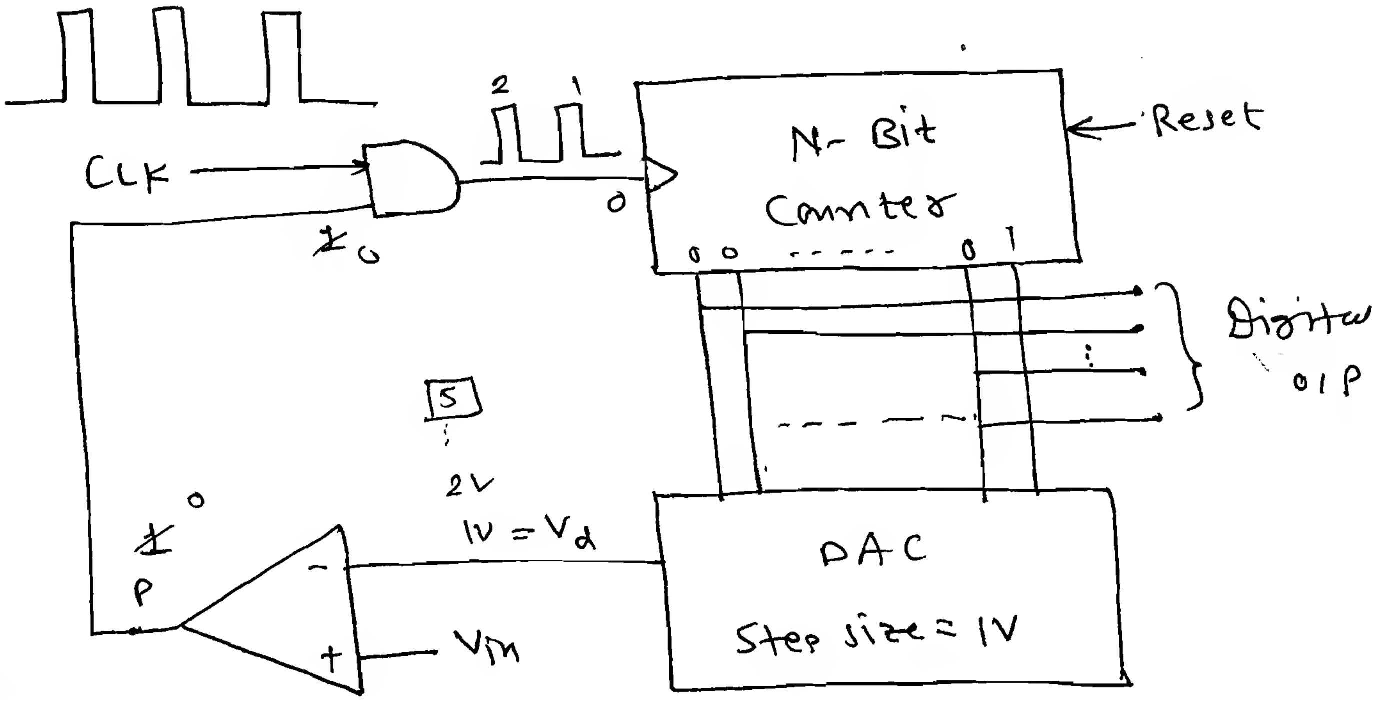
$= \left[\frac{1}{2} + \frac{1}{8}\right] \times 8.$

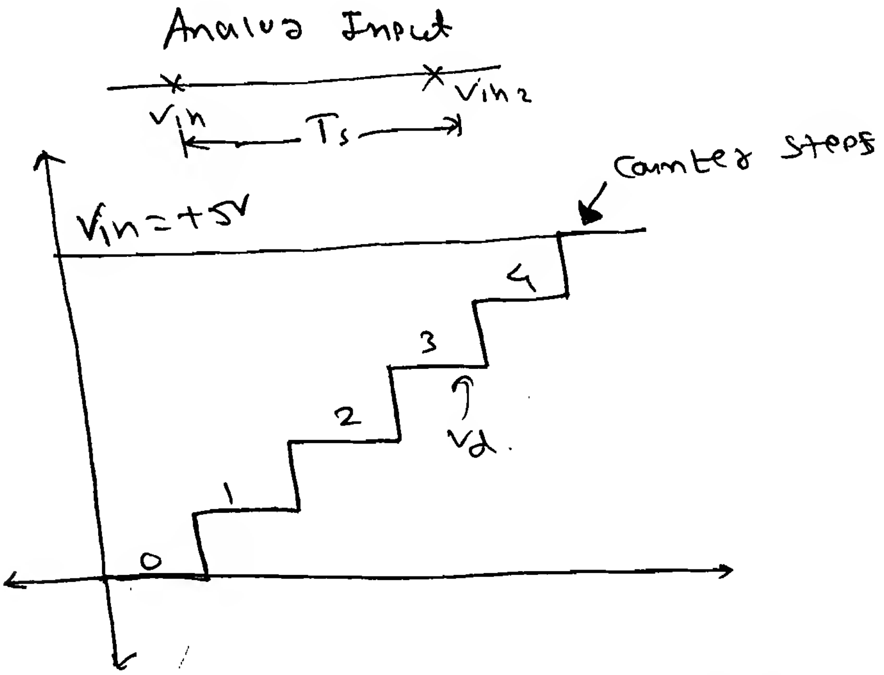
$= \cancel{8}\ \frac{5}{8} \times 8$

$\therefore \boxed{V_0 = 5V}$

* Analog to Digital Converter:

(1) Counter Type ADC:





→ Value of Counter = ceiling value ~~Voltage~~ $\left(\frac{\text{Input voltage}}{\text{step size}} \right)$.

* Two Features

① Conversion time depends on the input magnitude.

② Maximum Conversion time

$= \underline{\underline{2^N - 1.}}$

$$\left.\begin{array}{l} 000\cdots\cdots 00 \\ \vdots \\ \vdots \\ 20V \rightarrow 111\cdots\cdots 11 \end{array}\right\} \begin{array}{l}(2^N-1) \\ \text{CLK pulshes.}\end{array}$$

Sampiing period :

$T_s \geq$ max Cnv. time.

i.e $\boxed{T_s \geq (2^N-1)\,T}$

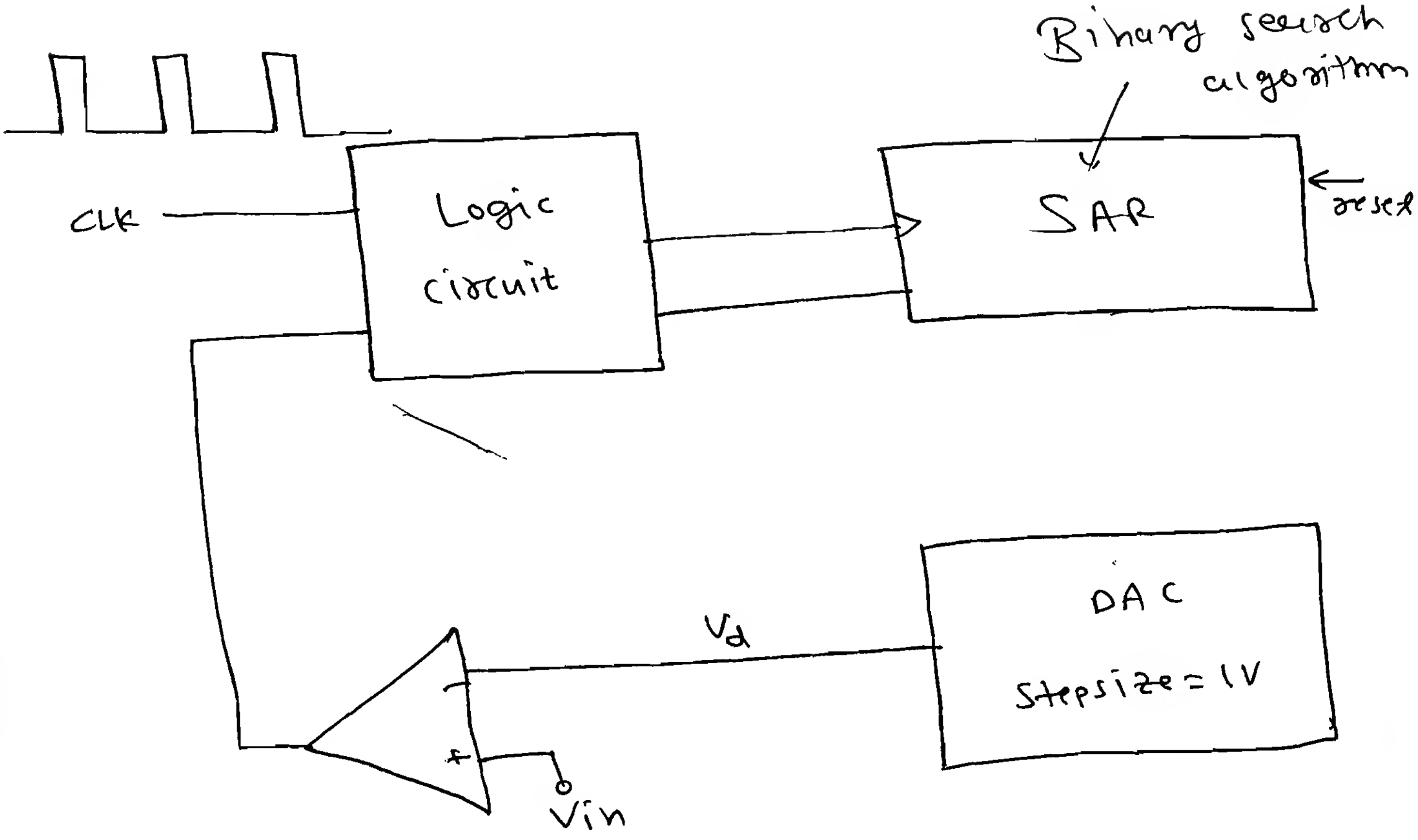
$\downarrow \qquad \downarrow$

Sampiig Rate $f_s\uparrow = \dfrac{1}{T_s\downarrow} = \dfrac{1}{(2^N-1)\,T}$. Hz.

NOTE:

→ In Counter type A to D converter the conversion time doubles for every 1 bit increase in size.

② Successive Approxi ADC:



→ $V_d > V_{in} \Rightarrow V_d \downarrow \Rightarrow SAR \downarrow$.

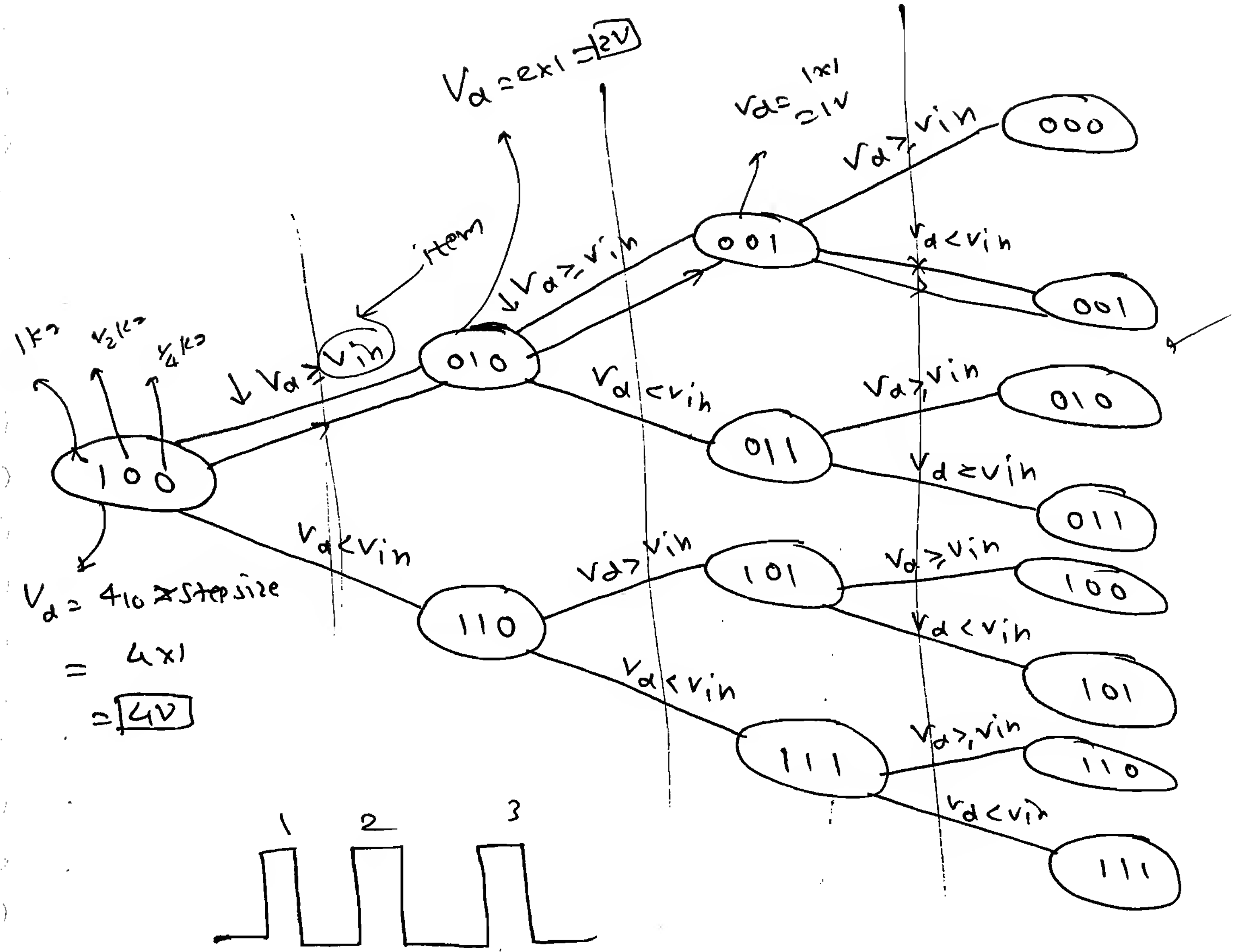
→ $V_d < V_{in} \Rightarrow V_d \uparrow \Rightarrow SAR \uparrow$.

$V_d > V_{in} \Rightarrow P = 0.$

$V_d = V_{in} \Rightarrow P = 0$

$V_d < V_{in} \Rightarrow P = 1.$

* 3 - Bit SAR



$\boxed{\text{Time} = 3T.}$

→ In successive approximation A to D Converter the digital O/P is always less than the analog input.

P) $V_{in} = 2V.$ $V_d = 4V.$

DAC stepsize = 1V.

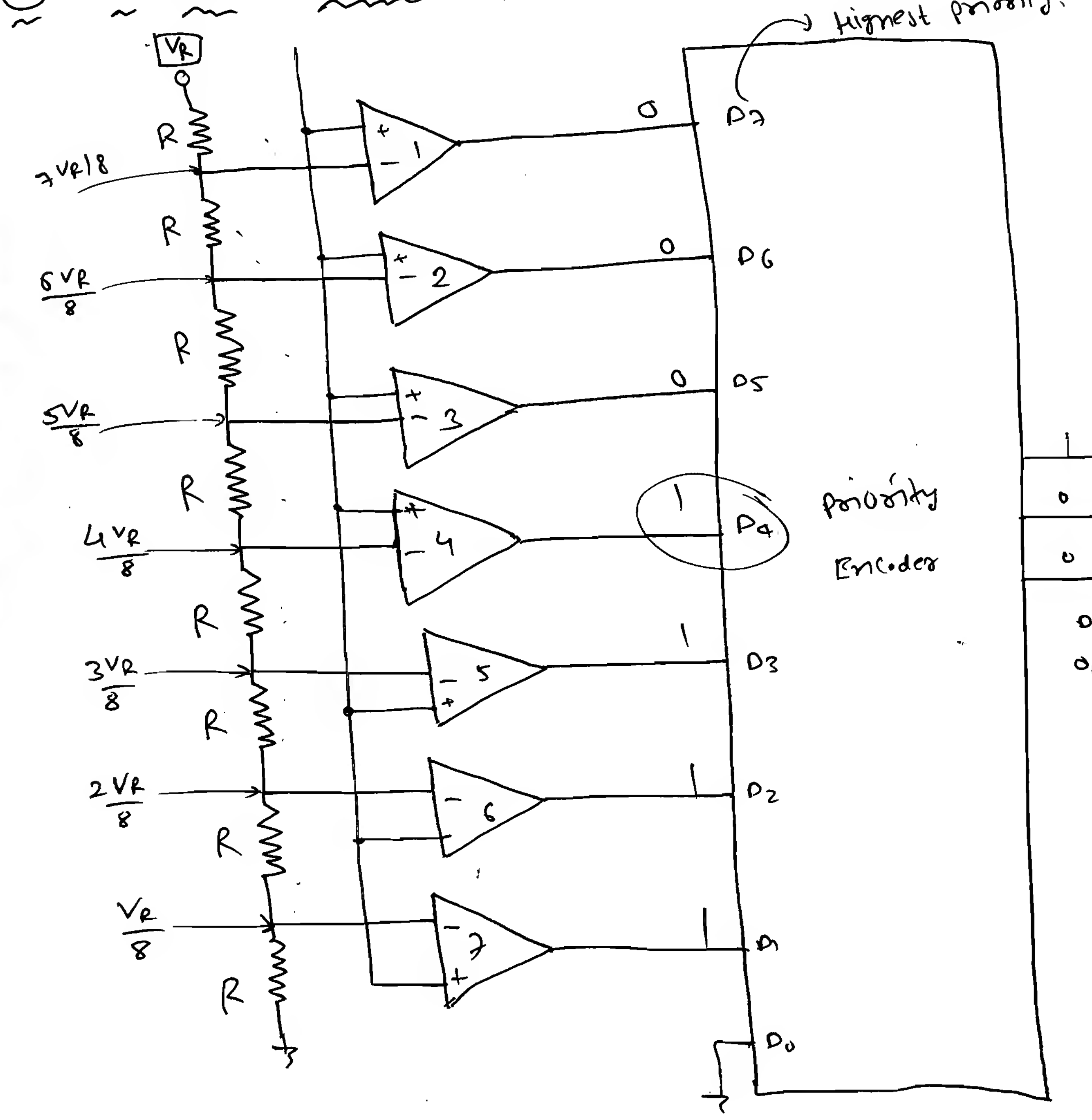
Dig output = $\underline{001.}$

→ In successive approximation ADC,

① conversion time doesn't depend on input magnitude.

② Maximum conversion time = NT.

③ 3-Bit Flash (parallel) ADC:



Vin	D7	D6	D5	D4	D3	D2	D1	D0	outputs A	B	C
$\frac{4V_R}{8} \le Vin \le \frac{5V_R}{8}$	0	0	0	1	1	1	1	0	1	0	0
$\frac{2V_R}{8} \le Vin \le \frac{3V_R}{8}$	0	0	0	0	0	1	1	0	0	1	0

→ In flash time ADC

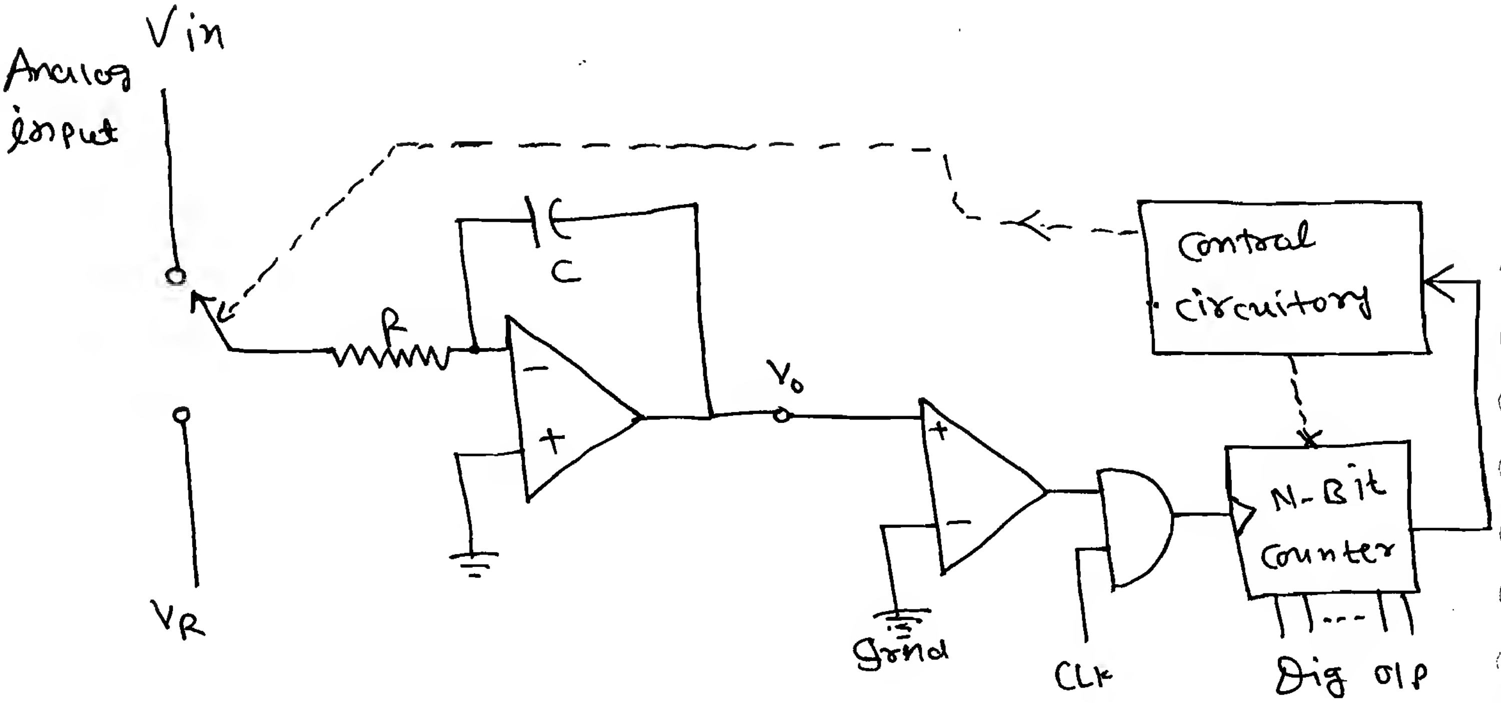
1. Conversion time doesn't depend on input magnitude.

2. Maximum conversion time is very less it deem and hence it is the fastest ADC.

* Disadvantage

→ N-Bit flash ADC requires $(2^N - 1)$ Comparators.

(4) Dual Slope (or) Integrating ADC:



* Vin, VR Polarities must be opposite.

1) $V_o > 0$ => CLCK Pulses reach the Counter.

~~2)~~ $V_o \leq 0$ => Counter Stops.

2) $V_o = \frac{-1}{RC} \int v\,dt$.

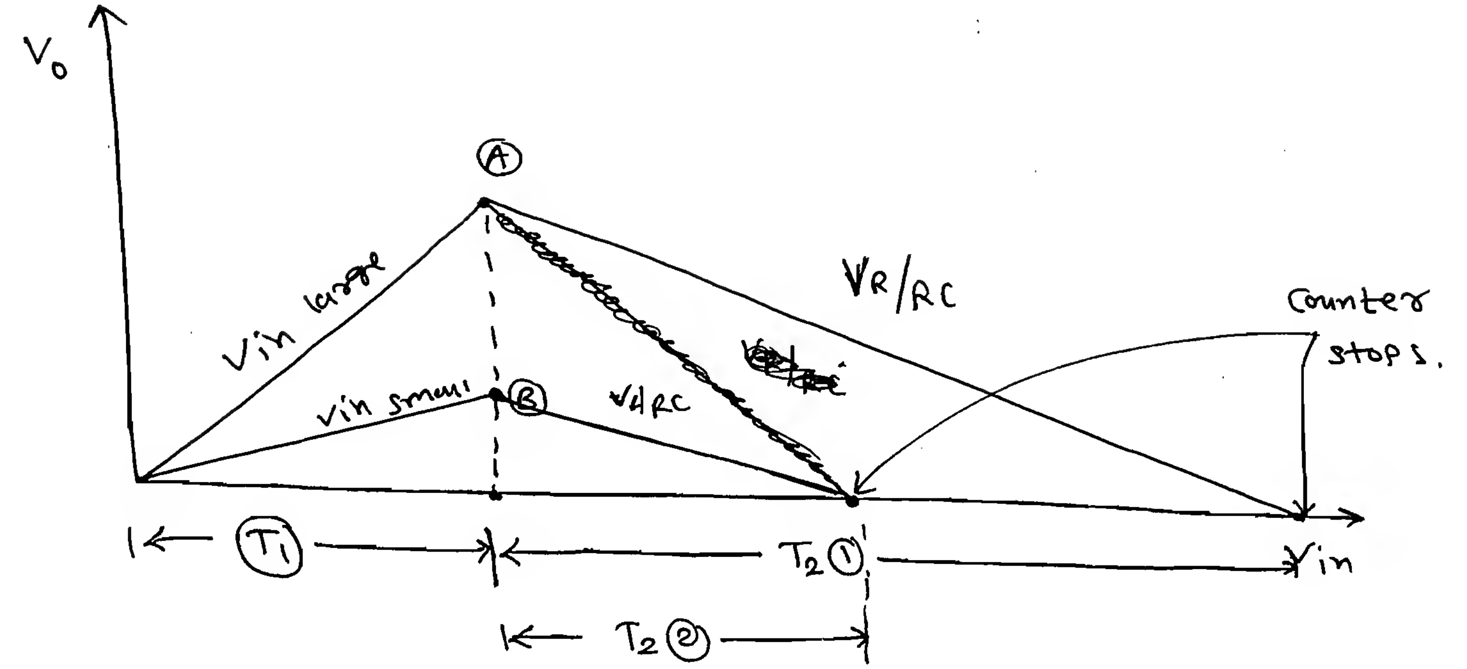
If V is constant

$\therefore \boxed{V_o = -\frac{V}{RC} \cdot t}$

3) Control circuitary.

→ For fixed time "T_1" => { "S → Vin" and Counter is Reset

(b) After "T_1" $\Rightarrow$ $\begin{cases} \text{"} S \longrightarrow V_R \text{"} \text{ and} \\ \text{Counter is started.} \end{cases}$



CLOCK: $\Rightarrow T = 2\ \mu s$.

T_2 ① : 100µs. → Counter value: $\frac{100}{2} = 50$, 50 CP require

T_2 ② : 20µs. → Counter value: $\frac{20}{2} = 10$, 20 CP require.

If clock Period = 2µs.

Case-(i) Value of Counter $= \frac{100}{2} = 50 = (000\cdots110010)_2$.

Case-(ii) Value of Counter $= \frac{20}{2} = 10$
$= (000\cdots1010)_2$.

* Discharging time "T_2".

Voltage change during T_1 = Voltage change during "T_2".

i.e. $\frac{V_{in}}{RC} \cdot T_1 = \frac{V_R}{RC} \cdot T_2$

$$* \therefore \boxed{T_2 = \frac{|V_{in}|\, T_1}{|V_R|}}.$$

* Advantages:

① It is very accurate because the same capacitor is used for charging and discharging. ~~so~~, that if any deviation exist the system will not be affected. Hence, it is used in all digital voltmeters.

② The integrator at the input eliminates the ~~after~~ effect of power supply noise called as "50 Hz hum" (interference).

* Disadvantage:

→ It is speed of operation is very less.

In,
⇒ Dual slope ADC,

(i) Conversion time depends on input magnitude.

(ii) Max. conversion time

$$\therefore \boxed{T_{max} = T_1 + (2^N - 1)\, T}$$

N = size of ADC (No. of bits).

T = CP.

T_1 = fixed time.

But, $T_{1,max} = (2^N - 1)\,T.$

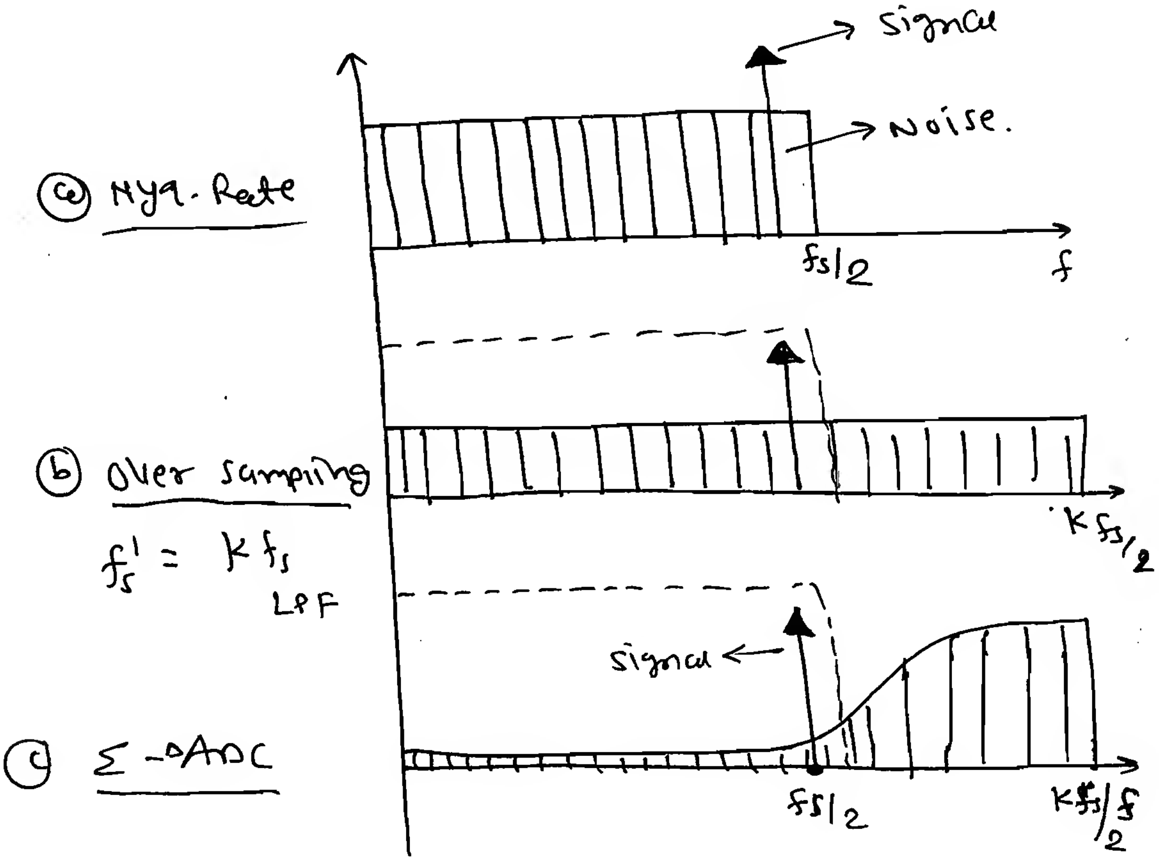
$$\therefore \; T_{max} = (2^N - 1)\,T + (2^N - 1)\,T$$

$$\therefore \; \boxed{T_{max} \approx 2^{N+1} \cdot T.}$$

→ In Dual slop ADC the conversion time doubles for every 1 bit increase in size.

5) Σ - Δ ADC (sigma - Delta ADC).
↳ 1 Bit DPCM.

Concept:



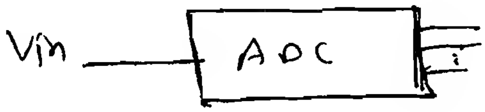
In Σ - Δ ADC

signal → LPF $\left(\frac{1}{s+1}\right)$.

Noise → HPF $\left(\frac{s}{s+1}\right)$.

* Resolution of ADC:

→ It is the minimum Chunge required at the input to obtained 1-Bit Change at the output.



$0V \longrightarrow 000 --- 000$

$10mV \longrightarrow 000 \cdots 00.$

Resolution $\boxed{15mV} \longrightarrow 000 ----01.$

$$\text{Resolution} = \frac{\text{Input Voltege Range}}{(2^N-1)}.$$

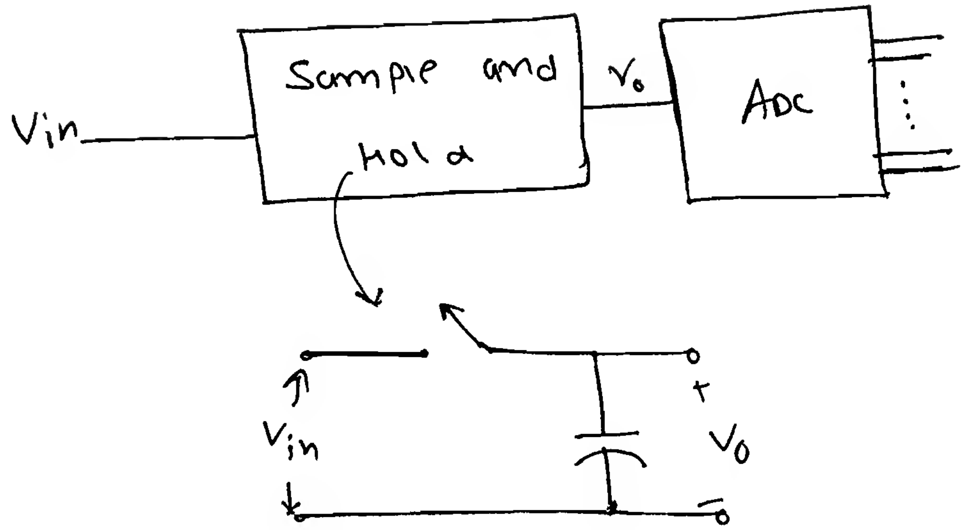
* For Proper ADC operation.

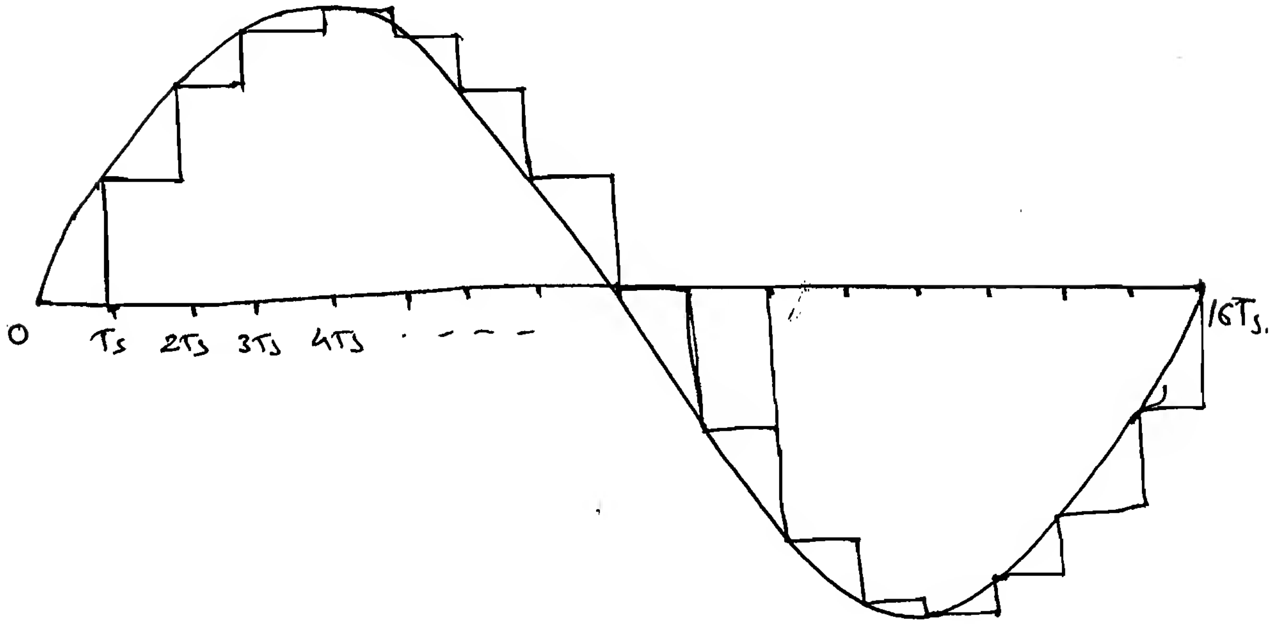
⇒ Ripple Voltuge at Input < Resolution of ADC.

* Continous signal can't be Apply to the ADC. ADC can not convert continous analog signal into digital signal directly. we require some circuity for that. i.e. Sample and Hold Ckt.

*

→





→ Acquisition time:

→ It is the time taken for the switch to close and capacitor charge to the input voltage.

→ Aperture time:

→ It is the time taken by the capacitor to disconnect from i/p after the switch is open.

Ex-9 Page-29 CR

→ $T_1 = 300ms$ $V_R = 100mv$

$T_2 = 370.2ms$

$\therefore |V_{in}| T_1 = |V_R| T_2$

$\therefore V_{in} = \frac{370.2}{300} \times 100mv$

$$\boxed{V_{in} = 123.4 . mv}$$

Q-10

Hint: In ADC,

$\text{Quantization error} = \pm \frac{\text{stepsize}}{2}$

$\text{Stepsize} = \frac{1.275}{2^8 - 1} = 5 mv.$

$\therefore \text{Quantization error} = \pm \frac{5}{2}$

$= \pm 2.5 mV.$

Q-11 Maximum Conversion time

$T_{max} = 2^{N+1} . T$

$= 2^{11} . 1 \mu s$

$T_{max} = 2048 \mu s.$

$\therefore T_s \geq T_{max}.$

$\therefore T_s = 2048 \mu s.$

$\therefore f_s \leq \frac{1}{2048 \mu s}$

$\therefore \boxed{f_s \leq 500\,Hz.}$

Let, $\boxed{f_s = 500\,Hz}$

$\therefore$ Input $BW = \frac{f_s}{2}$

$= \frac{500}{2}$

$\therefore \boxed{BW = 250\,Hz.}$

Q-5

$FSO = (2^N - 1) \times Stepsize.$

$\therefore 20 = (2^8 - 1) \times Stepsize.$

$\therefore Stepsize = \frac{20}{255}$ volts.

$\therefore V_{in} = (219)_{10} \times \frac{20}{255}$

$\therefore \boxed{V_{in} = 17.V}$

Q-3 8-bit ADC → 0-5V.

0V ⟶ 0000 0000 = 00H.

⋮

2.5V ⟶ 1000 0000 = 80H

⋮

5V ⟶ 1111 1111 = FFH.

2) 8-bit ADC → -5 to +5

-5 ⟶ 0000 0000 = 00h.

⋮

0 ⟶ 1000 0000 = 80h

⋮

5 ⟶ 1111 1111 = FFH.

MICROPROCESSORS

→ 80% Instruction Set

10% Mem, I/O Interfacing

10% Miscelleneous.

→ Topics:

1. Block diag. & pin diag of 8085 µP.
2. Mem Interfacing.
3. I/O Interfacing.
4. Instruction, Machine cycles.
5. Timing Diagram.
6. Instruction set.
7. Addressing modes.

$\boxed{8085 \ \mu P}$

~~1) 16 Address~~ * Features:

① **16 Address Lines**

$$\begin{tikzcd}
 & A_{15} - A_8 \text{ (Higher order Addr. lines)} \\
\boxed{A_{15} - A_{\cancel{1}0}} \arrow[ur] \arrow[dr] & \\
 & A_7 - A_0 \text{ (Lower order Addr. lines).}
\end{tikzcd}$$

$$\text{Mem. Capacity} = 2^{16} = 2^{6} \cdot 2^{10} = 2^{6} \cdot 1\text{kB}.$$

$$= \boxed{64 \text{ kB.}}$$

② **8 Data lines:**

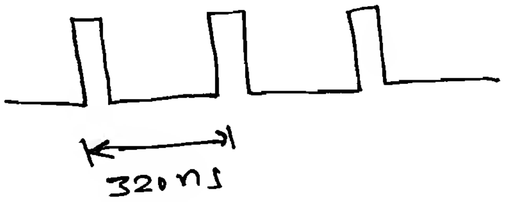
$$= \boxed{D_7 - D_0}$$

③ **Frequency of μP.**

$f = 3.072$ MHz

$\boxed{f = 3.072 \text{ MHz}}$; Clock period $T = 1/f$

$$\boxed{T = \frac{1}{f} = 320\text{ns.}}$$



④ ' **Von Neumann' Architecture.**

↓

X [Harvard Architecture].

→ In Von Neuemann architecture Programm and data are store into the same memory.

→ In Harward Architecture Program and data are stored in separate memories with separate Buses.

Eg: ARM microcontroller.
Dsp processors.

⑤ 'CISC' Procecessor.

CISC = Complex instruction set Computer.

↓

X (RISC) = Reduced instruction set computer.

e.g. (i) ARM Controller
(ii) Dsp processor.

⑥ NMos Tech.

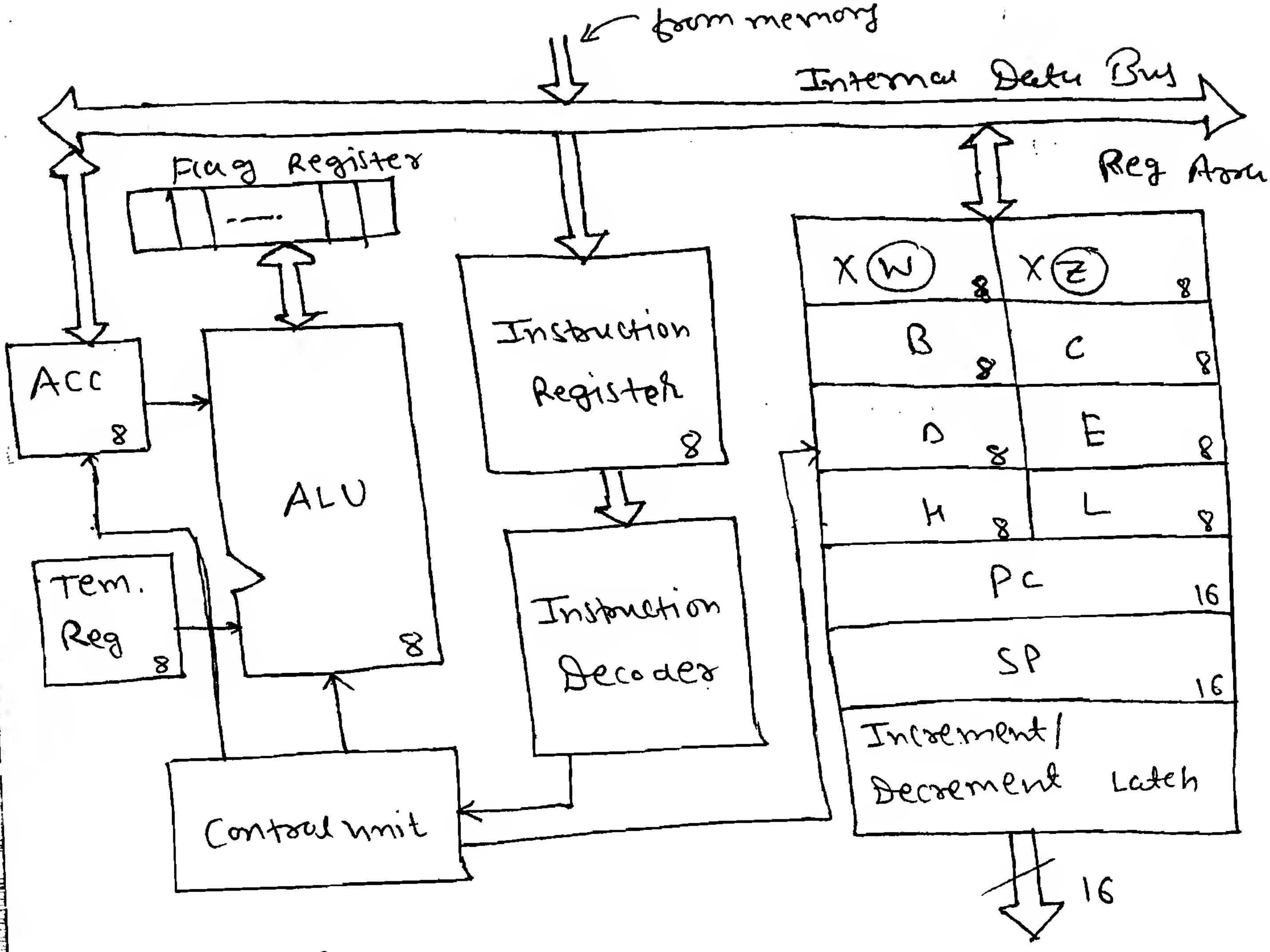
NOTE:
4004, 8008
used +5, -5, +12
supply.

⑦ Only +5V Supply is used.

⑧ TTL Compatible.

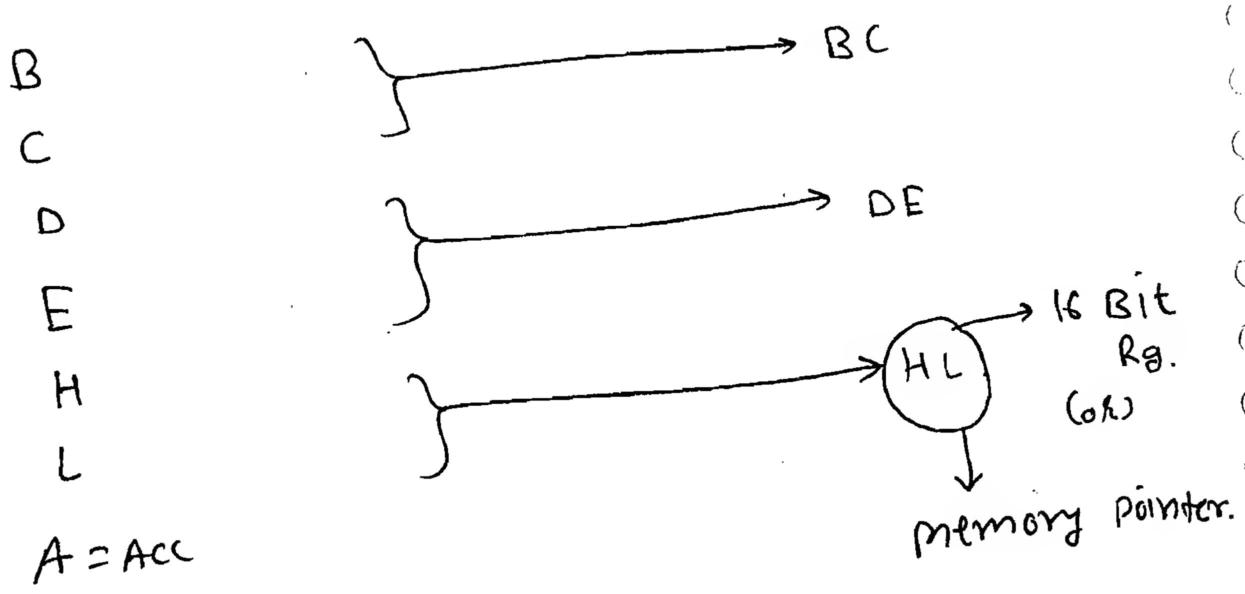
⑨ 8085 µp is a 8-bit µp because its ALU capacity is 8-bits.

* Block Diagram of 8085 µP.



W, z = Temp. Reg.

→ 8-bit Register (R) => (7). , Reg. pairs | 16 bit Reg. (RP) => (3)



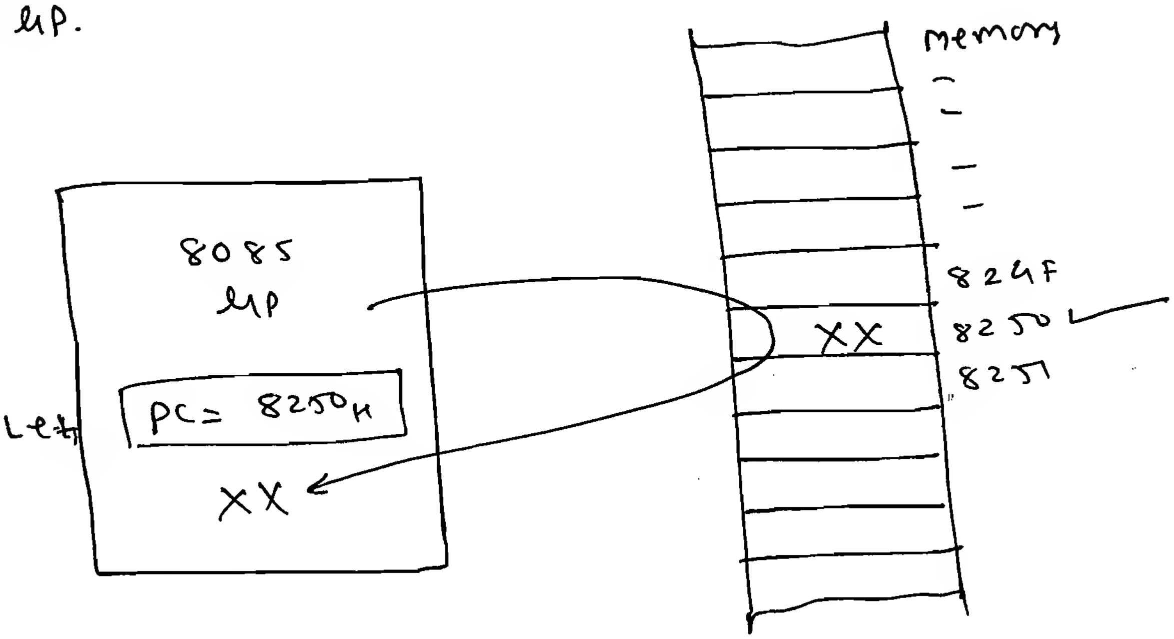
→ PSW_{16} = Program Status Word_{16}

$= \boxed{ACC_8 \;\big|\; \text{Flag Register}_8}$

→ PC: Program Counter.

⇒ Program Counter contains the address of the next instruction.

→ It is called IP (Instruction pointer) in 8086 µP.



⇛ SP_{16} = Stack Pointer_{16}

* Flag Register:

⇒

S	Z	X	AC	X	P	X	CY

S: Sign ~~b~~flag.

Z: zero flag.

~~A~~p: Parity flag.

cy: Carry flag.

① Sign flag:

→ S=1 if MSB of ALU Result =1

⇒ S=1 if ALU Result is '-ve'.

② zero flag:

→ z=1 if ALU operation Resuts = 00_H.

③ Parity flag.

→ P=1 if ALU Resuts has "even parity".

④ CY: Carry Flag:

→ cy=1 if carry occurs during Addition (OR) borrow occurs during Subtraction.

⑤ Ac: Aux Carry:

→ AC=1 if Carry occurs from Lower to Upper Nibble (or) if borrow occurs from Upper to Lower Nibble.

NOTE:

→ The Programmer can not excess the AC flag, μP uses internally for BCD addition.

E.g.

$$\begin{array}{rcrr} & & y=1 \quad x=1 & AC=1 \\ & & ~~x=1~~ & \\ ED_H & = & 1110 & 1101_2 \\ +\ CB_H & = & +\ 1100 & 1011_2 \\ & & 1 \quad\quad 1 & 11\ \ 1 \\ \hline & & \underline{\underline{1}}011 & 1000 \end{array}$$

$S = 1$

$Z = 0.$

AC=1.

$P = 1$ (even Parity).

$CY = 1.$

* As $x = 1$, $y = 1.$

⇒ <u>No overflow</u>

(or)

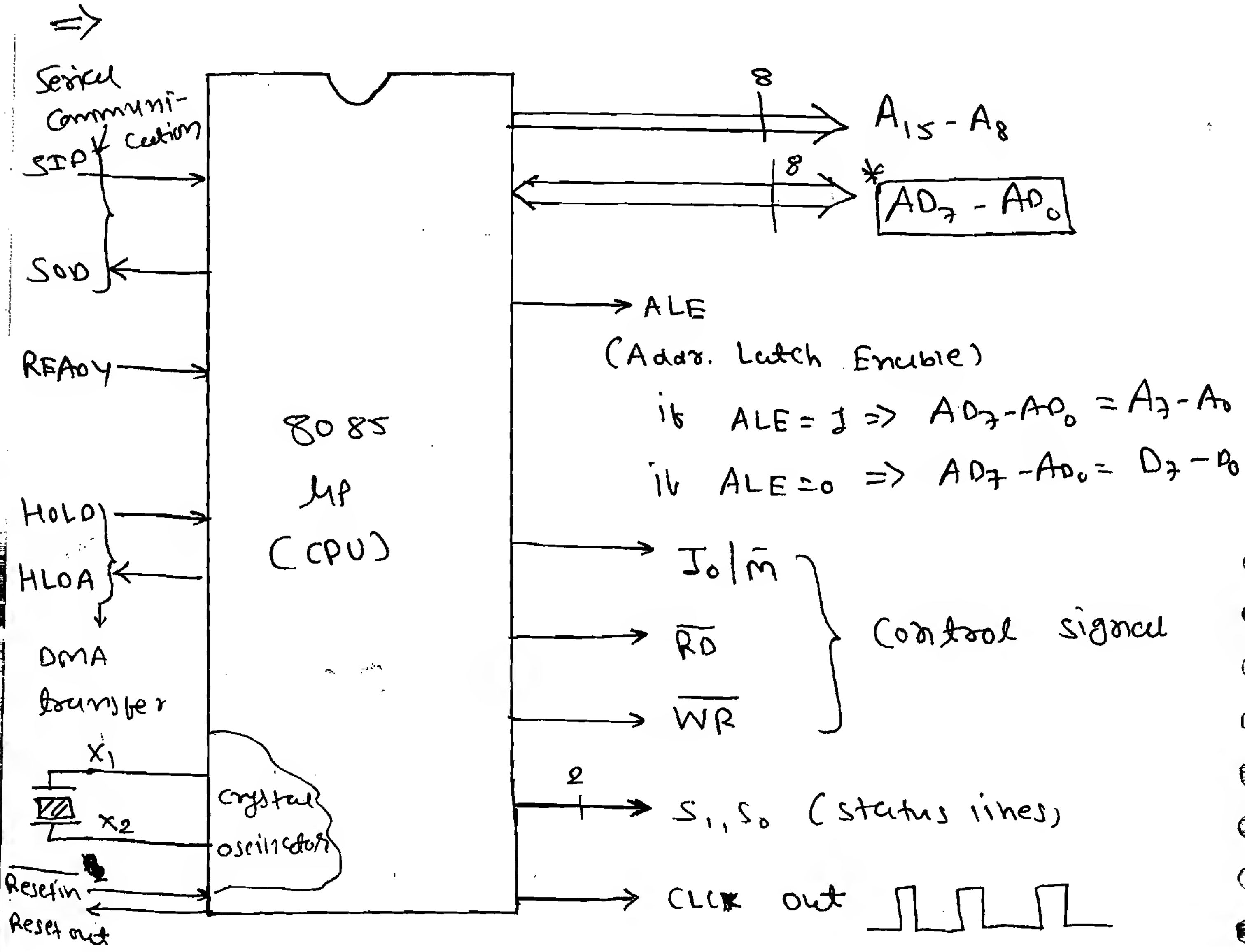
* Two −ve Nos are added.

Result is −ve

⇒ No overflow.

* Pin Diagram of 8085 µP.

=>



=>

$IO/\overline{M}$	S_1	S_0	operation
0	1	1	opcode Fetch M/C
0	1	0	mem Read M/C
0	0	1	mem write M/C
1	1	0	~~PA~~ IO Read M/C
1	0	1	IO write M/C
1	1	1	Interupt Ack M/C

=> Multiplexing:

$A_8 - A_{15} \longrightarrow 8$

$\left.\begin{matrix} A_0 - A_7 \\ D_0 - D_7 \end{matrix}\right\} \boxed{AD_0 - AD_7} \rightarrow 8$

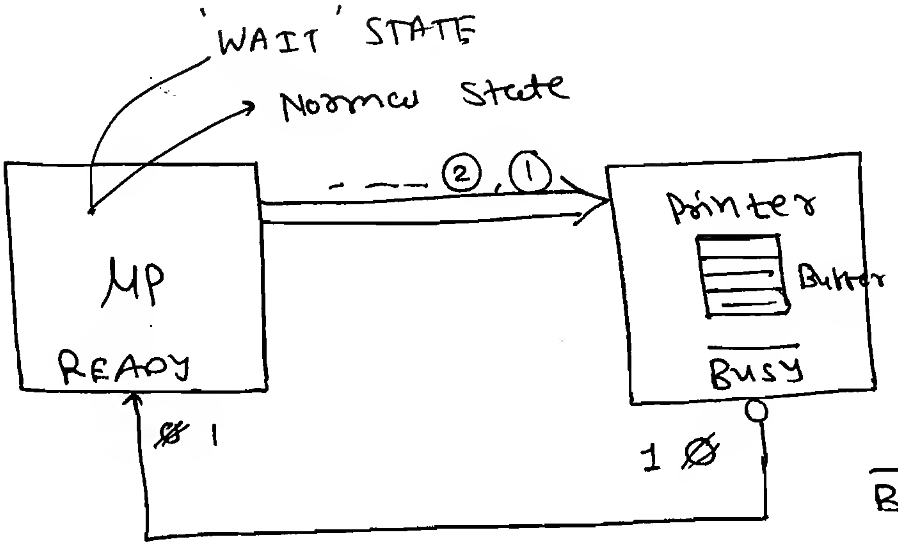
= 16.

* READY

{ 1 Normal
{ '0' WAIT State

→ Ready pin is used to synchronised the µp with slow speed peripherals and memories.

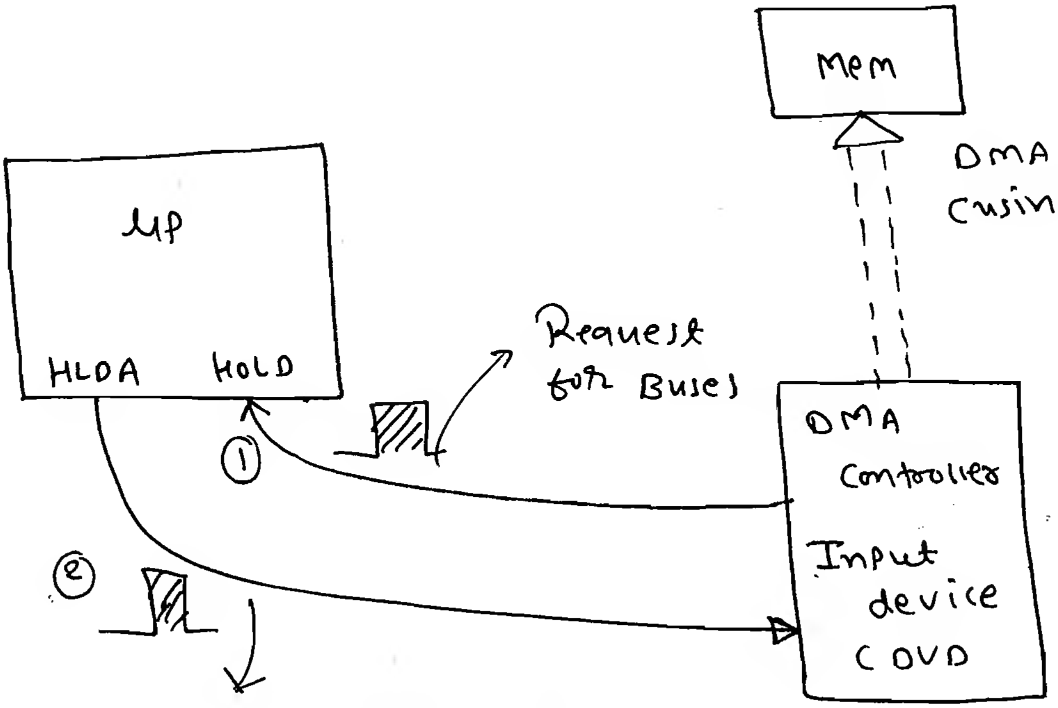
Eg.



$\overline{\text{BSY}}$ = 0 printing
= 1 over.

* HOLD and HLDA:

e.g.



DMA transfer (using 3 Buses).

Buses are given to DMA Controller.

DMA = Direct memory excess.

→ In DMA transfer μp acquires the Buses from DMA Controller when the HOLD pin becomes Low

→ Two Modes of operation:

ⓐ Burst mode:

→ In this mode the Buses are handover to the microprocessor only after the entire data transfered to memory.

ⓑ Cycle Stealing:

→ In this mode the Control of the Buses Switches Back and forth bet^n μp and DMA Controller.

* X_1 & X_2:

→

$f = 3.072$ MHz.

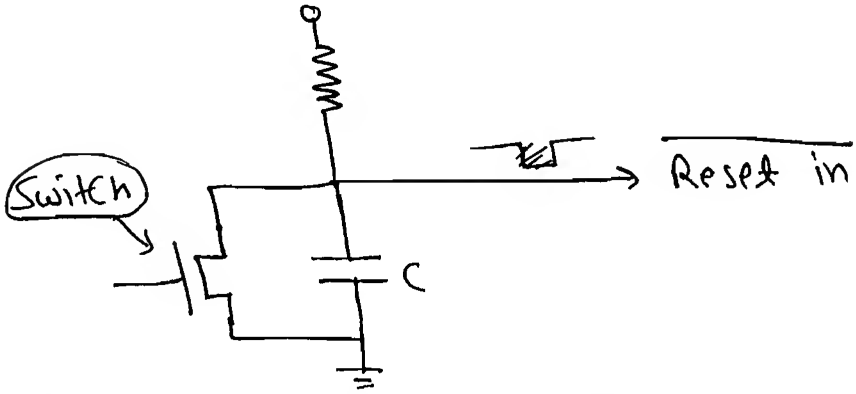
$$f_{crystal} = 2 \times f_{\mu p}$$
$$= 2 \times 3.072 \text{ MHz}$$
$$\boxed{f_{crystal} = 6.144 \text{ MHz}}$$

→ The Crystal freq. is divided by two to convert single phase clock into a two phase clock because 2 phase MOSFET Shift Registers are faster than single phase

MOSFET Shift Register.

* $\overline{\text{Rest in}}$ & Re Set out.

=> Power on Reset



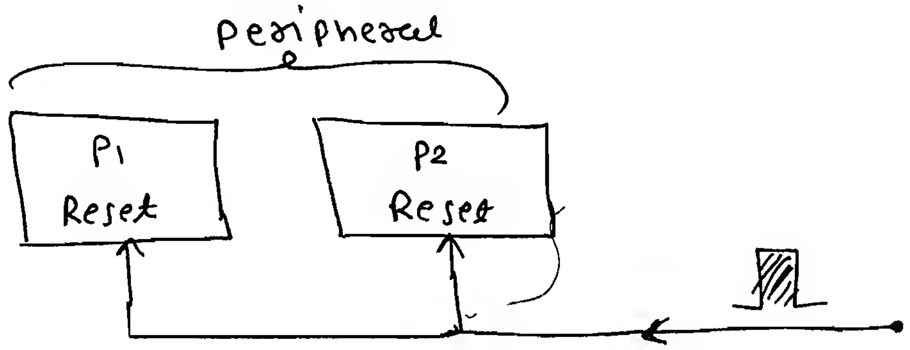
→ when up is reset

(i) All the Registers are clear including PC.

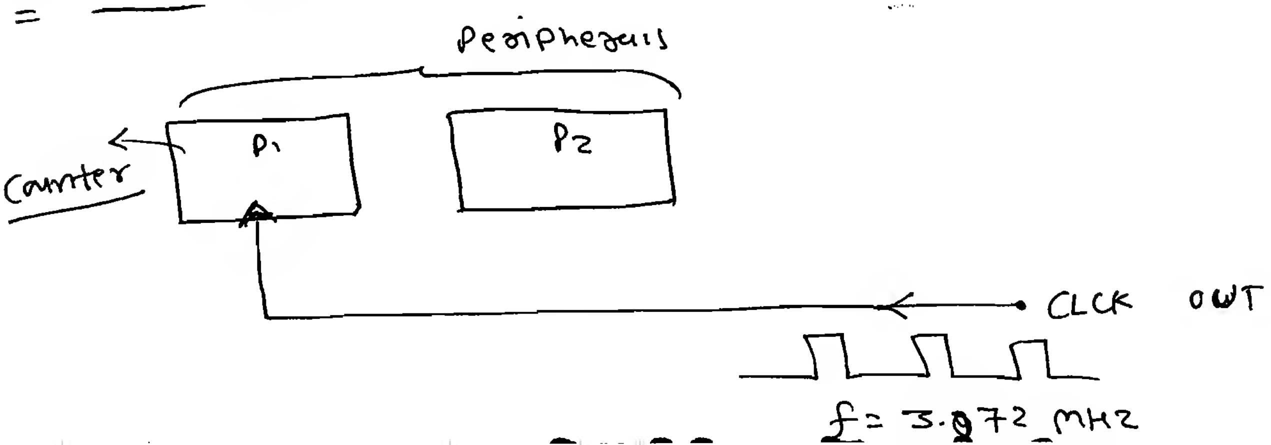
(ii) All the Buses entre into high impedence State.

(iii) up Processor Fetches its next instruction from memory location 0000H.

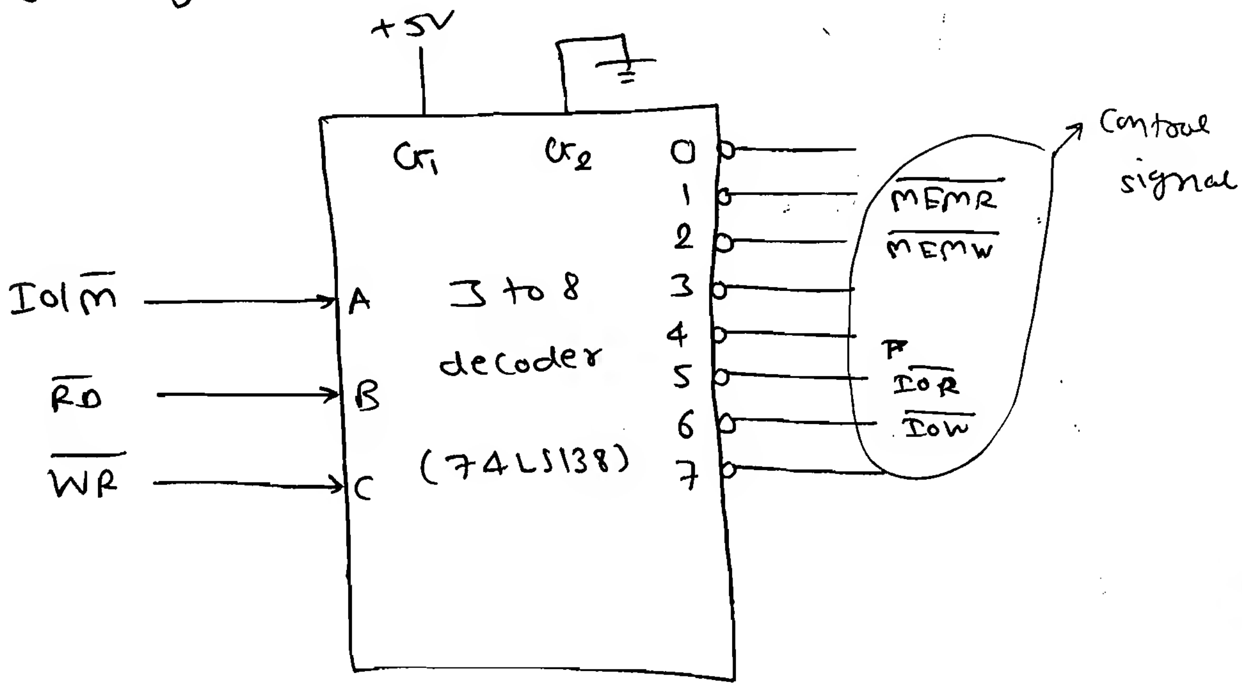
=>



* CLK out:



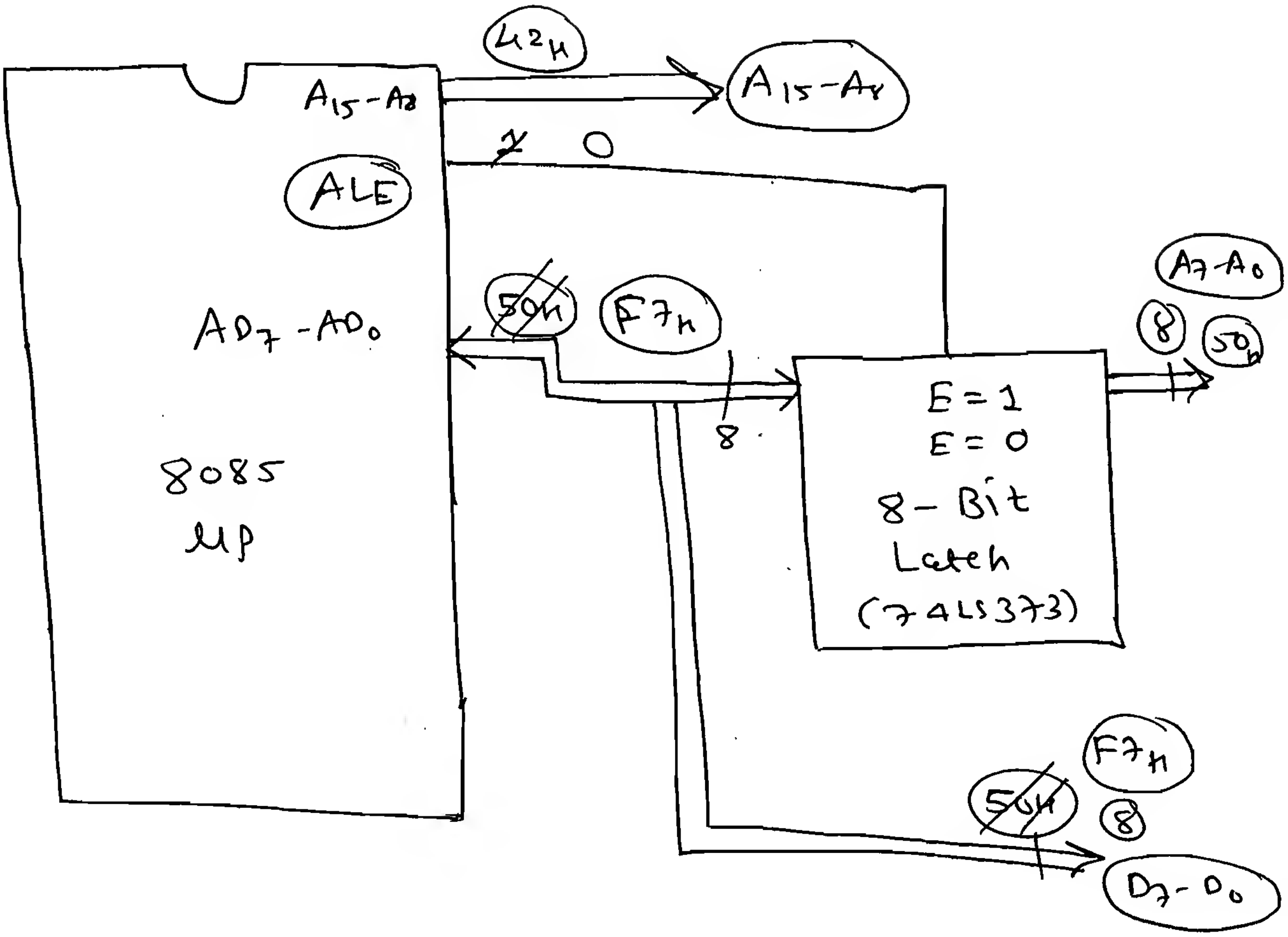
Ex: In the following decoder, determine the number of Vaid outputs and their functions



Ans:

$I_o/\overline{m}$	$\overline{RD}$	$\overline{WR}$	outputs
0	0	0	0 → Invalid
0	0	1	1 → $\overline{MEMR}$
0	1	0	2 → $\overline{MEMW}$
0	1	1	3 → Invalid
1	0	0	4 → Invalid
1	0	1	5 → $\overline{IOR}$
1	1	0	6 → $\overline{IOW}$
1	1	1	7 → Invalid.

* Demultiplexing of $AD_7 - AD_0$.



μP

① Address = 4250_h.

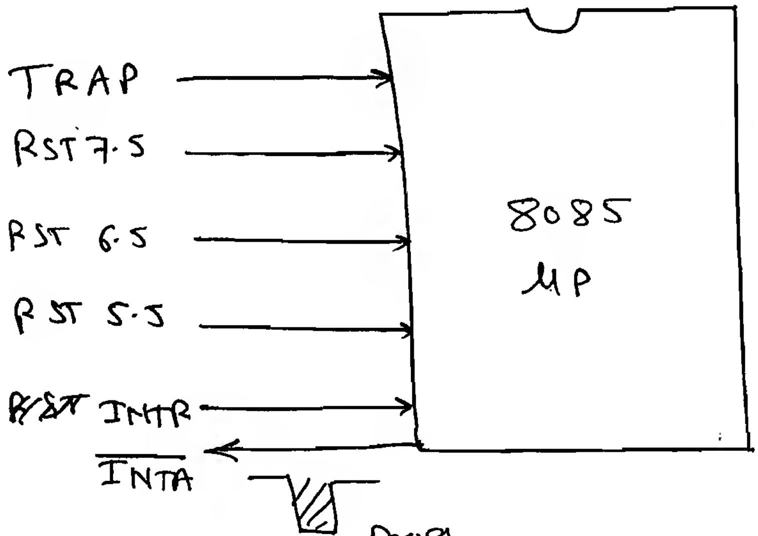
② Data = $F7_H$.

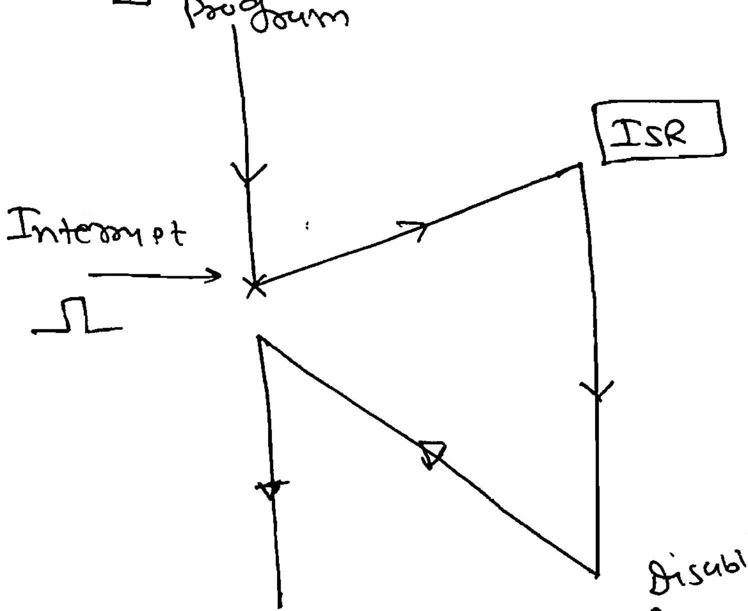
* Latch is enable, when Address is sent on $\boxed{AD_7 - AD_0}$

* Latch is disable, when data is sent on $\boxed{AD_7 - AD_0}$

→ The disadvantges of multiplexing is it reduces the μP speed of operation because of the time taken in Demultiplexing.

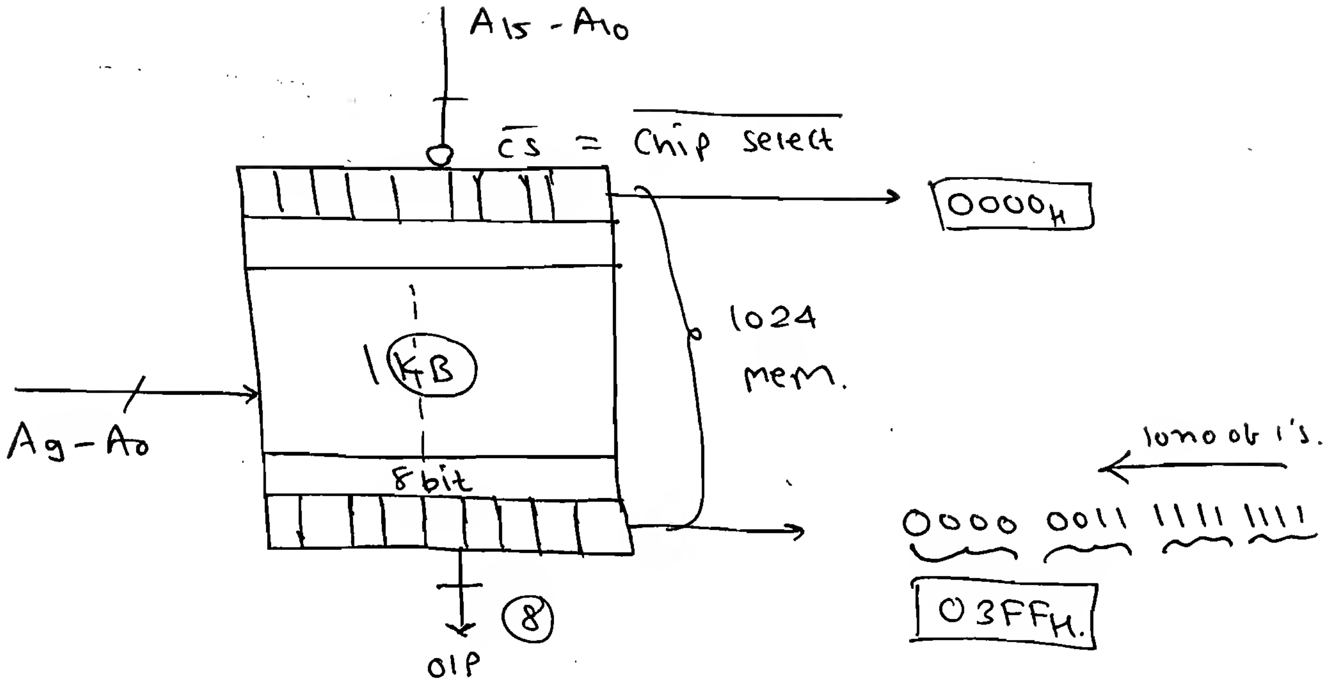
* Hardware Interrupts.





	Priority	Vector Address	Disabling ↑ Masking	Types of triggering
TRAP (RST 4.5)	Highest	0024H	NMI	Positive edge + Level
RST 7.5		003CH	Maskable	positive edge
RST 6.5		0034H		Level
RST 5.5		002CH		
INTR	Least	NVI		

* Memory IC's:



1 KB memory = 1024 × 8
= $2^{\boxed{10}}$ × $\boxed{8}$ (I/P, O/P)

→ if S.A = 3800 then E.A. = 3800 + 03FF = $3BFF_H$

→ if E.A. = FFFF then S.A. = $FFFF_H$ − $03FF_H$ = $FC00_H$

Ex-1 Determine the Starting address of 2764 memory IC if its ending address is $AA5F_H$.

Ans: **NOTE:**

MEM IC		RAM
(27) $\frac{16}{8}$ = 2KB (EPROM)	←	(6) $\frac{16}{8}$
	←	61 $\frac{32}{8}$
27 $\frac{32}{8}$ = 4KB		
27 $\frac{64}{8}$ = 8KB	←	61 $\frac{64}{8}$
27 $\frac{128}{8}$ = 16KB	←	61 $\frac{128}{8}$

→ memory size 8KB

$= 2^3 \times 2^{10} \times 8$

$= 2^{13} \times 8.$

13 → Address lines.

∴ $AA5F_H$.

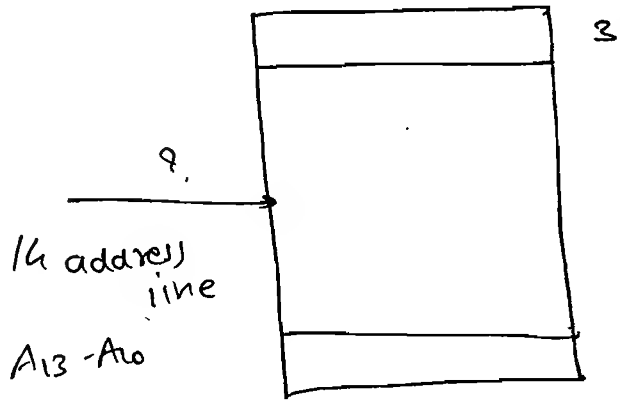
E.A. = 0001 1111 1111 1111

1FFF

∴ S.A. = $AA5F_H$
$- 1FFF_H$
$8A60_H$

Ex 2: Determine the size of the memory whose starting and ending address are $3A00_H$ & $79FF_H$ respectively.

Ans:



$3A00_H - 3A00_H = (0000)_H$

$79FF_H - 3A00_H = (3FFF)_H$

= 0011 1111 1111 1111

= 8

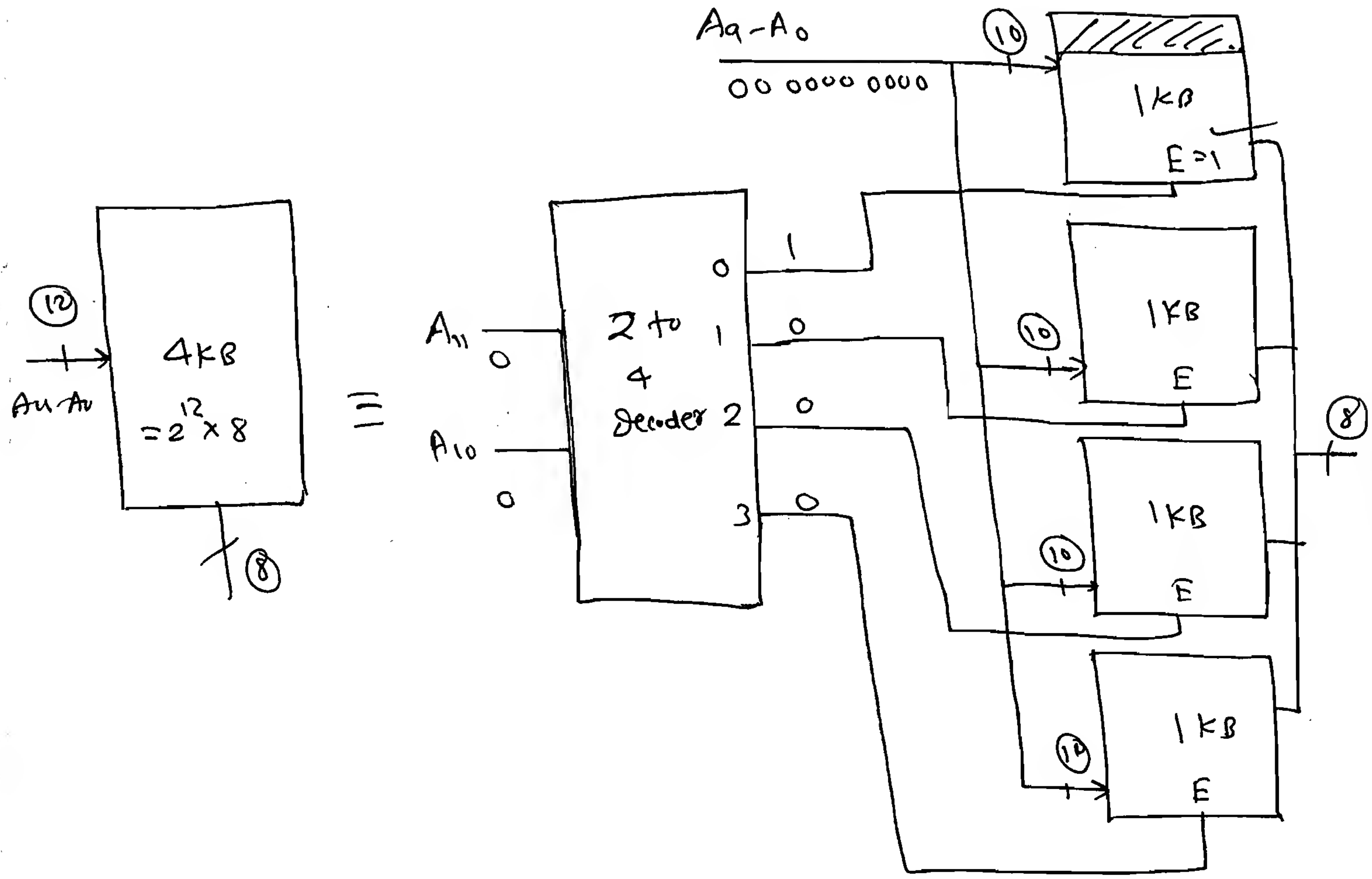
Hence, Memory size

$= 2^{14} \times 8 = 2^4 \times 2^{10} \times 8 = 16 \times 1KB = 16KB.$

* Memory Expansion:

* Construct a 4KB memory by using 1KB memory ICs

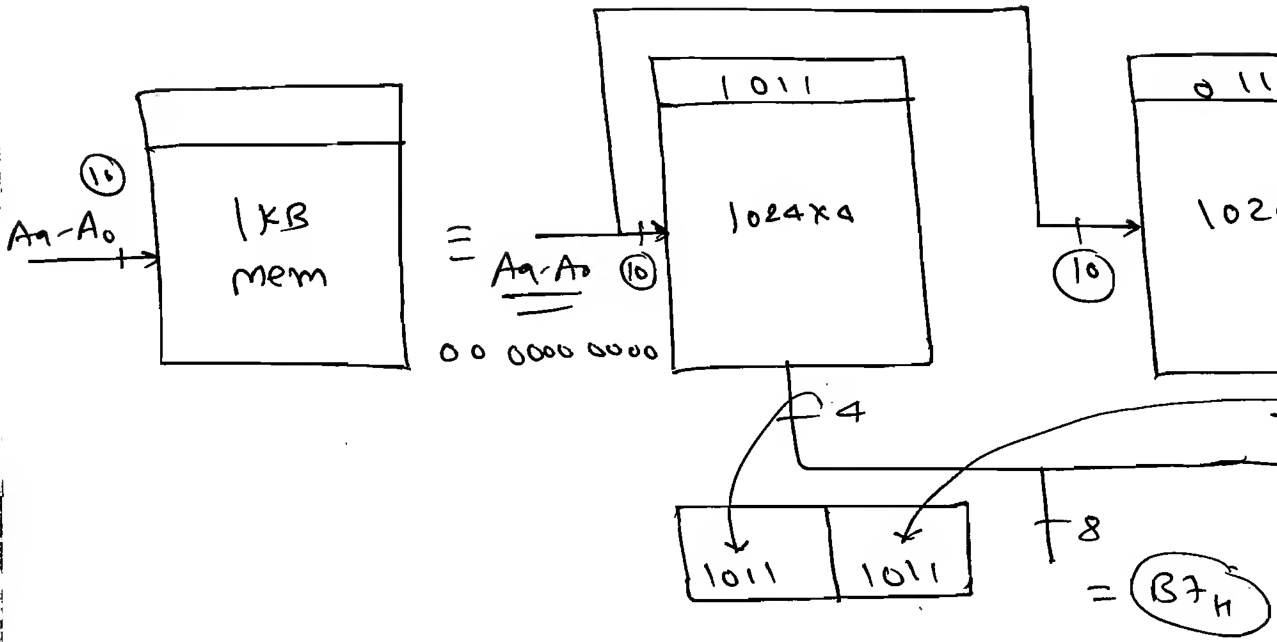
~~B 4 KB mem.~~



* Construct a 1KB memory using 1024x 4 mem IC's. → 1KB Mem. =1024 x 8.

P) Construct a 1KB memory using 1024×4 IC's.

→ 1KB mem = 1024 × 8.



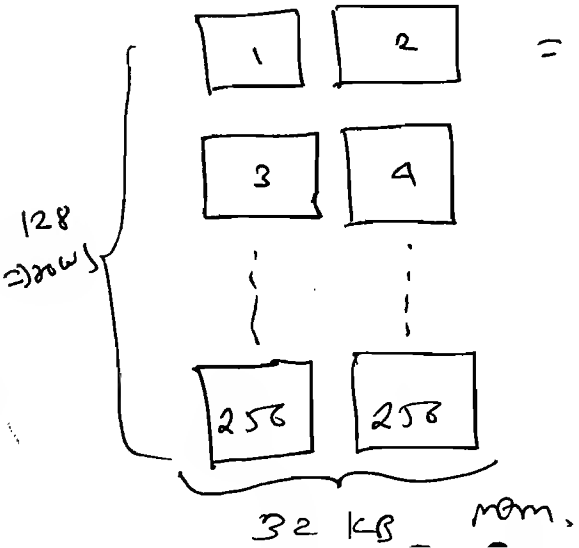
(P)

- How many 256 × 4 memory IC's are require. to construct a 32KB memory.

Ans: No. of mem ICs $= \dfrac{\text{Required size of mem}}{\text{Given size of mem.}}$

$$= \frac{32 \times 1024 \times 8}{256 \times 4.}$$

$= 256.$



Ex-2 A digital Computer has 20 address lines and 24 data lines How many memory ICs having 16 address lines and 8 data lines are required to feel the memory of the computer?

Ans:

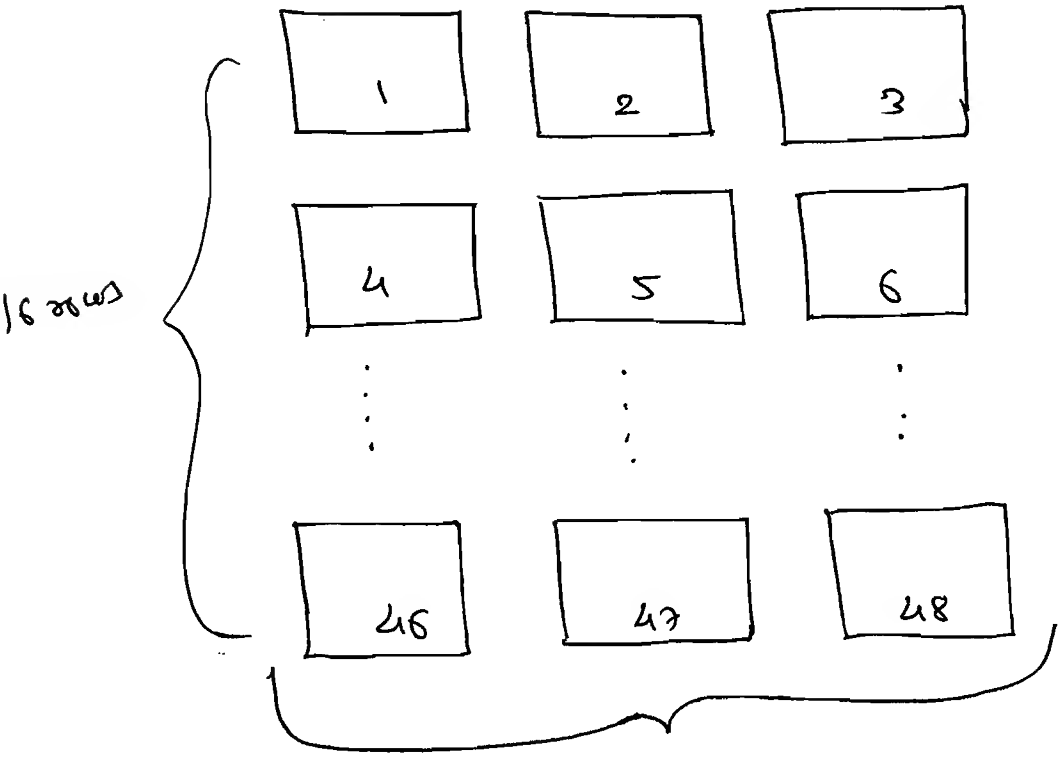
$$\text{No. of mem ICs} = \frac{\text{Required size of mem}}{\text{Given size of Mem.}}$$

$$= \frac{2^{20} \times 24}{2^{16} \times 8}$$

$$= 16 \times 3$$

$$= 48.$$

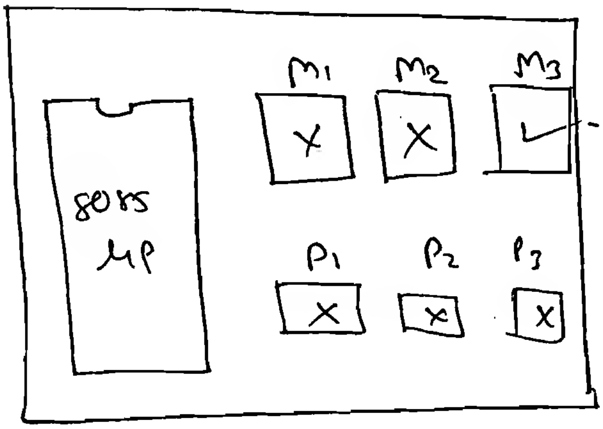
→ 16 rows, 3 columns.

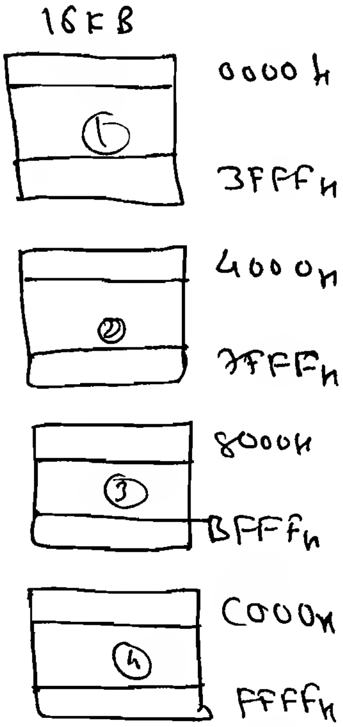


"$2^{20} \times 24$ = memory."

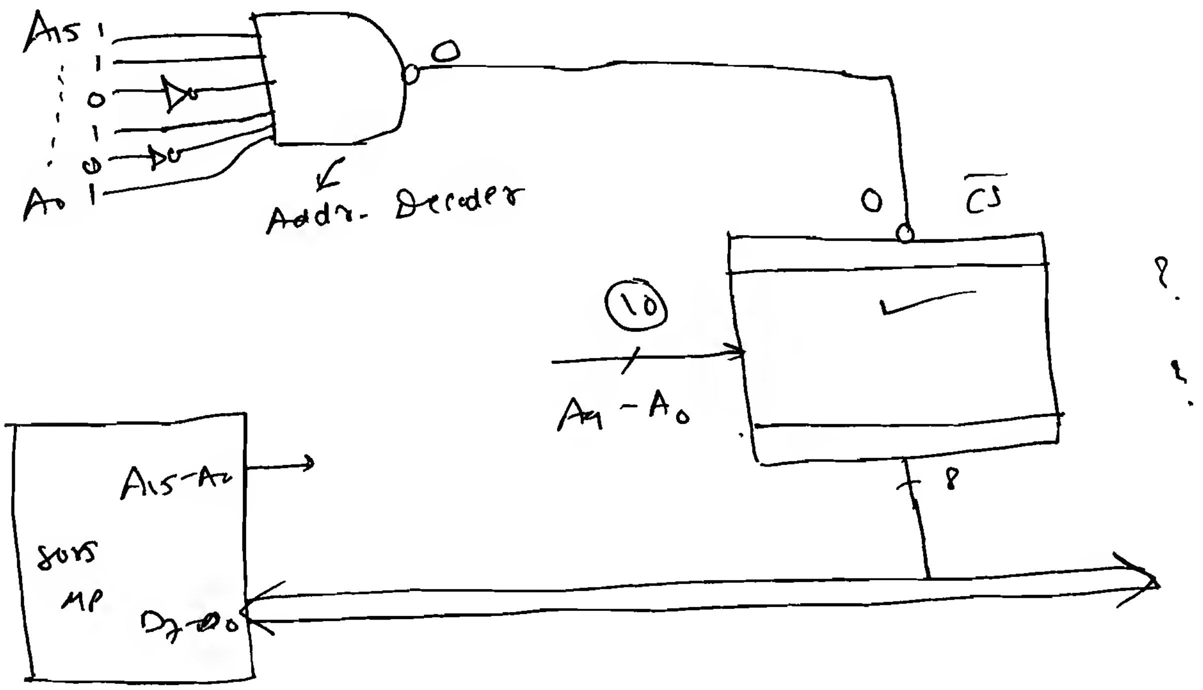
* Memory Mapping:

→ It is the Process of assigning address Range to all the memory ICs in a microcomputer.





Ex-1 In a micro Computer a 1KB memory IC is interface ~~with~~ inter the μP as shown in the following figure: Determine the address range of the memory IC



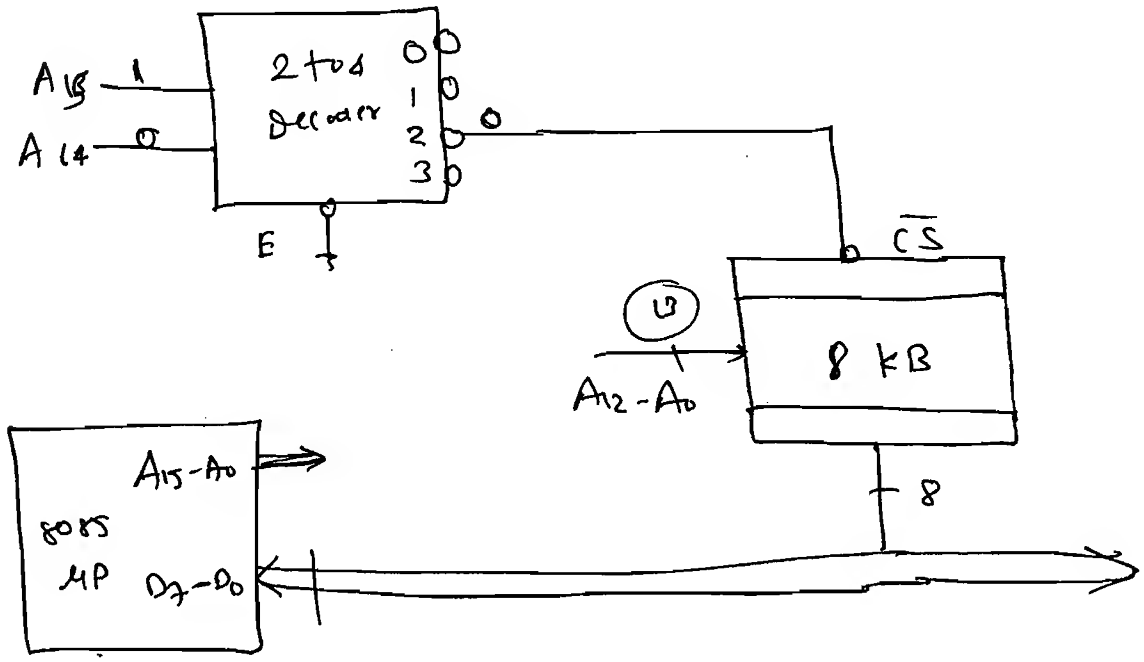
A_{15} ------ A_{10}	A_9 ------------ A_0	
1 1 0 1 01	00 0000 0000	= D400h
1 1 0 1 01	11 1111 1111	= D7FFh.

Addr. Range: D400H - D7FFh.

→ In the above interfacing ~~as~~ all 16 addr. lines are utilized it is called <u>absolute decoding</u> which resuts in one to one mapping.

<u>Ex-2</u> A 8kB memory IC is interface to the microprocessor as shown in the figure. Determine the address range of the memory IC.

<u>Ans</u>:



* A_{13} address line is unused.

A_{15}

Address decoding → unused

A_{15} - - - - - - A_{14}

→ Internal access.

A_{15}	A_{14}	A_{13}	A_{12} - - - - - - - A_0	x=0	x=1
1	0	X	0 0000 0000 0000 =	8000H	A000H.
⋮					
1	0	X	1 1111 1111 1111 =	9FFFH	BFFFH.

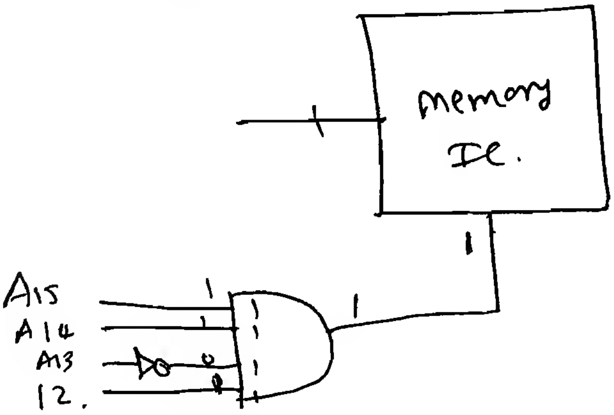
Fold back memory.

* "A_{13}" addr line is unused

⇒ Partial Decoding ⇒ many to one mapping.

NOTE: In partial decoding if $\underline{\underline{n}}$ address line are unused into results in $\underline{\underline{2^n}}$ fold back memories.

Ex 1 Which of the following indicate the address of the memory IC shown below:



→ mem is selected only when

$A_{15}A_{14}A_{13}A_{12}$ = 1101 = DH.

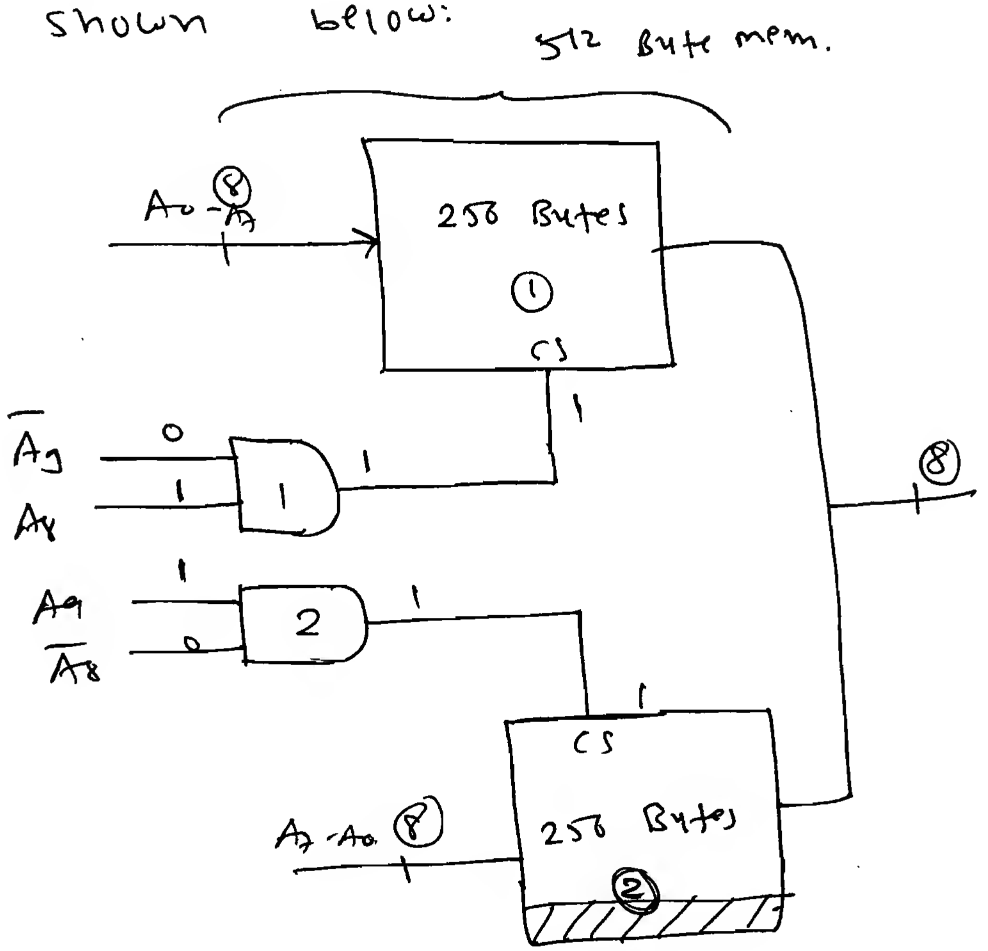
(a) B200H - B3FFH

(b) F800H - F9FFH

✓ (c) DA00H - DBFFH

(d) 8900H - 8AFFH

Ex-2. Which of the following doesn't indicate the address of the memory IC shown below:



$A_{15} - A_{10}$ A_9 A_8 $A_7 \ldots A_0$.

S.A. = 0 1 0000 0000.

E.A. = 1 0 0000 0000

→ Starting Add. must have $A_9 = 0$, $A_8 = 1$.

→ Ending Add. must have $A_9 = 1$, $A_8 = 0$.

A_{15} $A_9 A_8$ ~~$A_9 A_8$~~

ⓐ 3100H – 32FFH ⇒ 3100H = 0011 0001 0000 0000 valid

32FFH = 0011 0010 0000 0000

ⓑ 8D00H – 8EFFH ⇒ 8000H = 1101 valid

8EFFH = 1110

ⓒ F400H – F5FFH ⇒ F400 = 0100 Invalid

F5FFH = 0101

ⓓ C900H – CAFFH

⇒ C900 = 1001 valid.

CAFFF = 1010

☆ I/O Interfacing:

① Memory mapped I/O ⇒ I/O devices are treated as Memory.

② I/O mapped I/O ⇒ $\begin{cases} \rightarrow \text{mem} \\ \rightarrow \text{I/O.} \end{cases}$

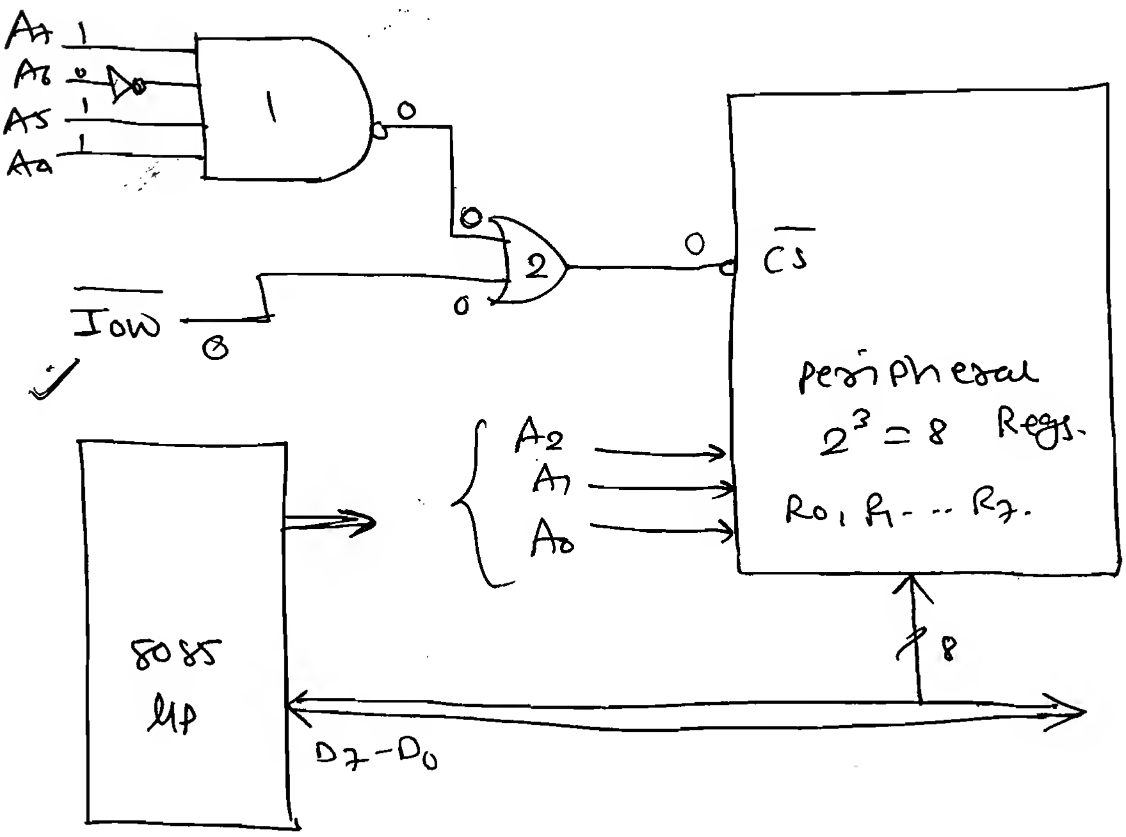
separate separate

	mem mapped I/O		I/O mapped I/O	
	mem	I/O	mem	I/O
① No. of Addr lines	16	16	16	8 (A_7-A_0)
② Control signals	$\overline{MEMR}$, $\overline{MEMW}$,	$\overline{MEMR}$, $\overline{MEMW}$,	$\overline{MEMR}$ $\overline{MEMW}$	$\overline{IOR}$, $\overline{IOW}$
③ No. of peripherals	* (i) — (ii) 60KB	64KB 4 KB	64KB +	$2^8 = 256$ I/O device.

*

	mem mapped I/O	I/O mapped I/O.
Advantages	1) $IO/\overline{M}$ pin is not required. 2) No separate instn for IO devices. 3) Arithmatic and Logic operations are directly performed on I/O ~~device~~ data.	1) Max capacity of μP is utilized. 2) Less Hardware Complexity for I/O interfacing.
Disadvantges:	1) more Hardware Complexity for I/O device interfacing 2) Effective memory space is reduced.	(1) Separate Instn for I/O device are required. (2) Can't perform Arith, Logical operations directly on I/O data.

Ex-1 A Peripheral is interfaced to the 85 µP as shown in the following figure. Determine (i) The mode of interfacing (ii) No. of internal Registers in the peripheral and their addresses.



Ans: 1) I/O mapped I/O Mode

(∵ only 8 addr lines used and Control signal is $\overline{IOW}$).

(2) As A_2, A_1, A_0 are Connected directly to the peripheral, no of Internal Registers

$= 2^3 = \boxed{8}$

~~A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0~~

Unused

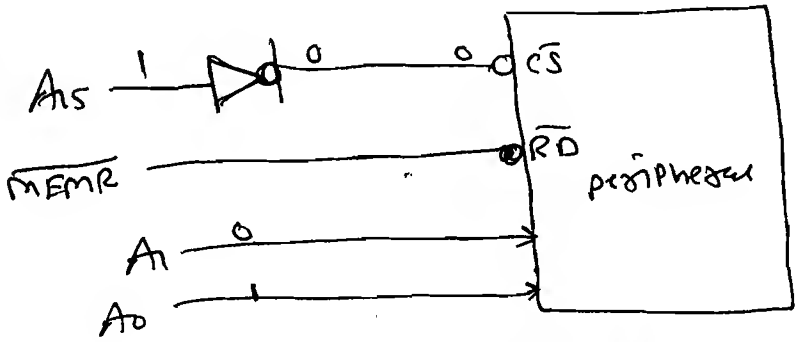
~~A7~~ ~~R~~

	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	$X=0$	$X=1$
R_0	ϕ	0	1	1	X	0	0	0	$B0_H$	$B8_h$
R_1	1	0	1	1	X	0	0	1	$B1_h$	$B9_h$
⋮	⋮					⋮			⋮	⋮
R_7	1	0	1	1	X	1	1	1	$B7_H$	BF_h

→ Address range: $B0_H$ - BF_H.

⇒ In IO mapped IO mode the value on higher order address mode $A_{15}-A_8$ is same as the value on Lower order add. Bus. (A_7-A_0).

Ex-2 Determine the max. addr. of Register B in the following peripheral!



Given ⇒

A_1	A_0	Register
0	0	A
0	1	B
1	0	C
1	1	D

Ans: memory mapped I/O.

→ For Register B.

A_{15}	A_{14}	----- - .		A_2 A_1 A_0
1	x xx	xxxx	xxxx	xxx 0 1.

→ Max Add. for Reg. 'B' is when all x's = 1.

1111 1111 1111 1101.

= $(FFFD)_H$.

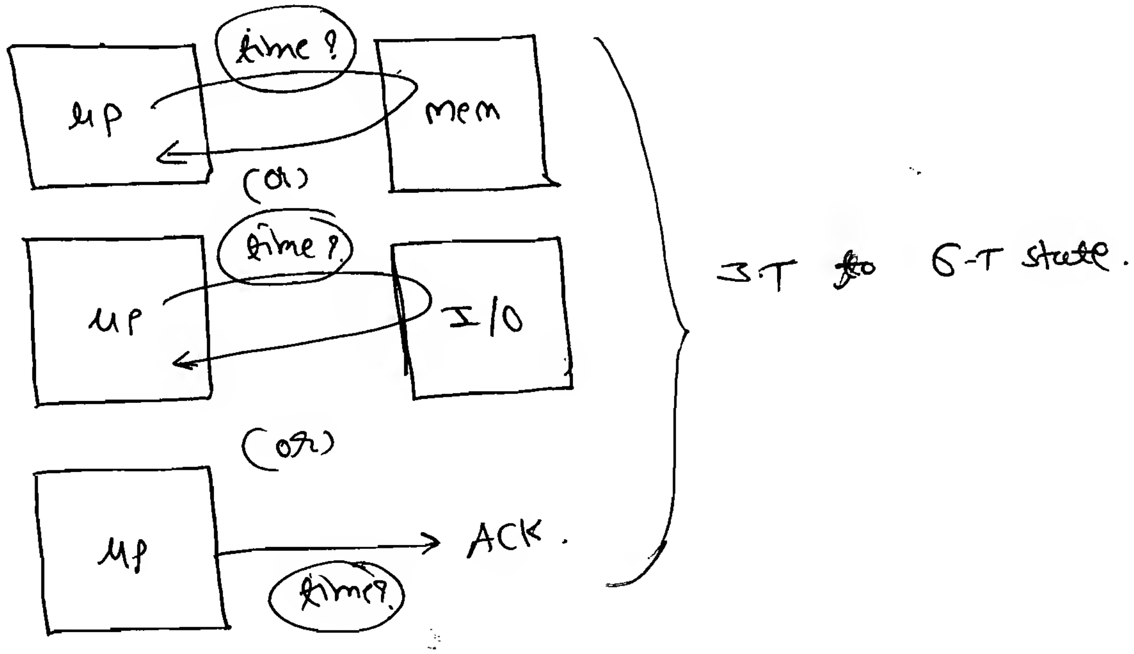
(1) Intruction cycle:

→ It is the time required to exerute an instruction

→ Range: 1 M/C to 5 M/C.

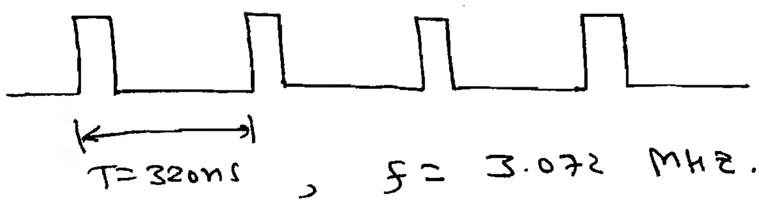
(2) Machine cycle:

→ It is the time required to Complete one operation of excessing memory, excessing I/o device or sending an acknoledgement.



③ T state:

→ It is the task performed in one clock period.



* Types of Machine cycle:

1) opcode fetch M/C (F) => $\boxed{4T} = 3T + \boxed{1T}$

⇓ time to decode the opcode.

2) mem. Read M/C (R).

3) mem. Write M/C (W).

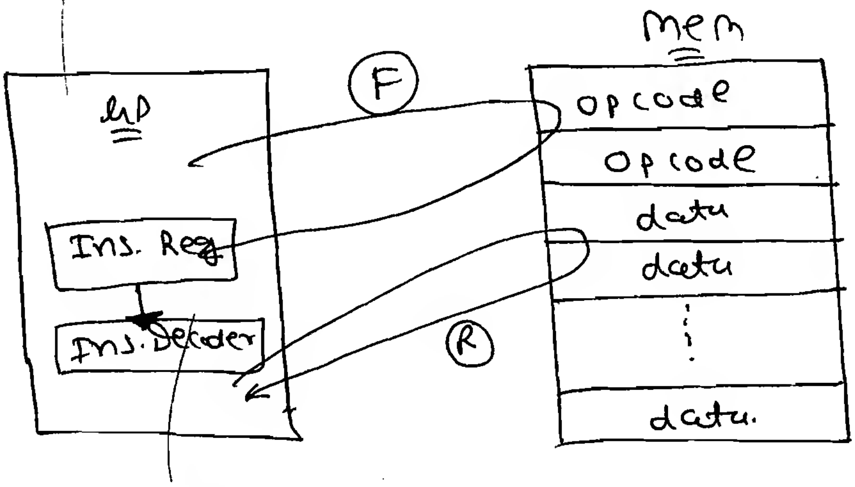
4) I/o Read M/C (I).

5) I/o write M/C (O).

} $\boxed{3T}$

6) Int. Ack M/C.

7) Hold Ack M/C.



=> Special M/C:

① S = opcode Fetch M/C (6T).

② B = Bus Idle M/C (3T).

*

ALP (Low Level)
(Assembly Languge Program)

8085 μp Hand coding

8085 μP Assembler

M/C Languge

HLL (C, C++, Java).

Compiler
Interpreter

M/C language

Eg:

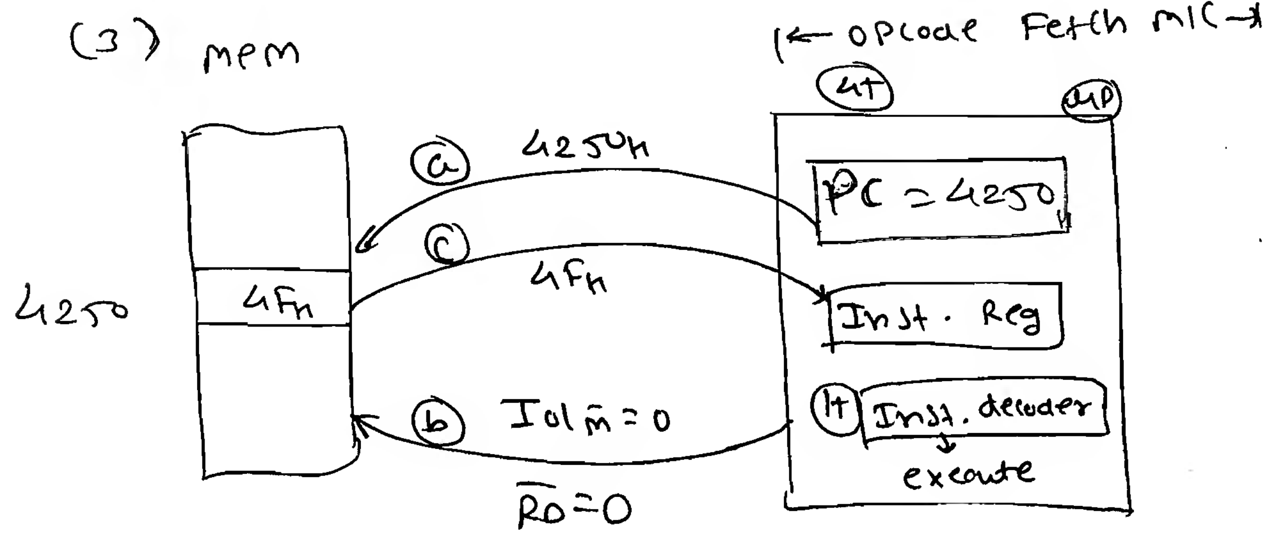
(1) Mov C, A ⇒ Ass. lang instruction.
(mnemonic).

Hand coding

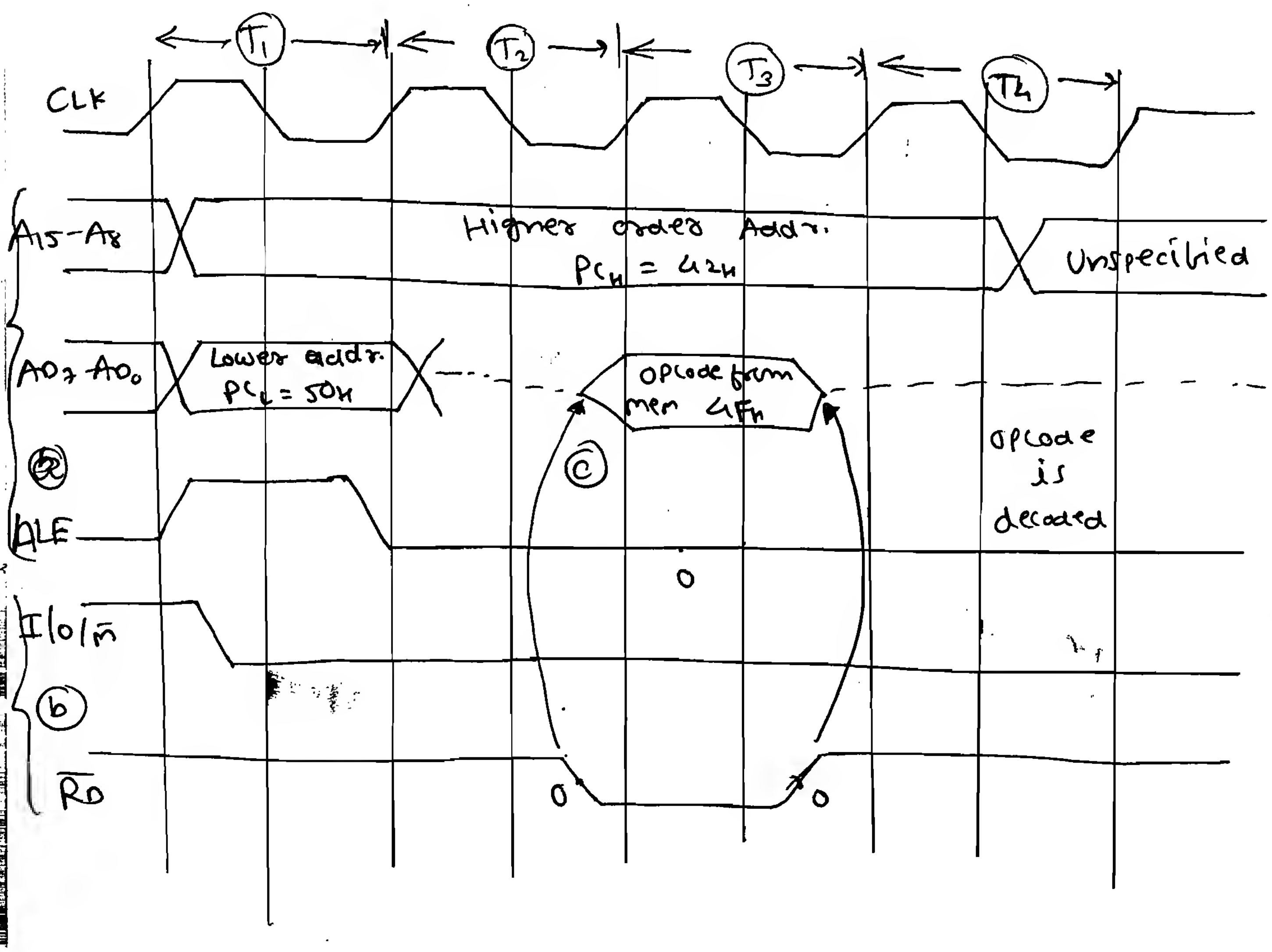
(2) Opcode = 0100 1111_2.
= $4F_H$.

⇓

(3) mem



* Timing Diagram of Opcode Fetch M/C:



→ $\boxed{PC_{16} = PC_H + PC_L}$ = 4250H

→ In T2 State of any machine cycle the PC is incremented

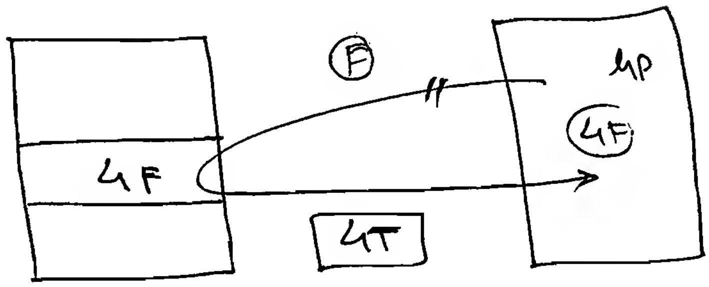
NOTE: In the Opcode fetch machine cycle if T4 State is removed then it converts into the timing diagram of memory Read M/C cycle.

* Types of Instructions (Based on size):

1. 1- Byte Instructions.
2. 2- Byte Instructions.
3. 3- Byte Instructions.

⇒ ① 1- Byte Instruction:

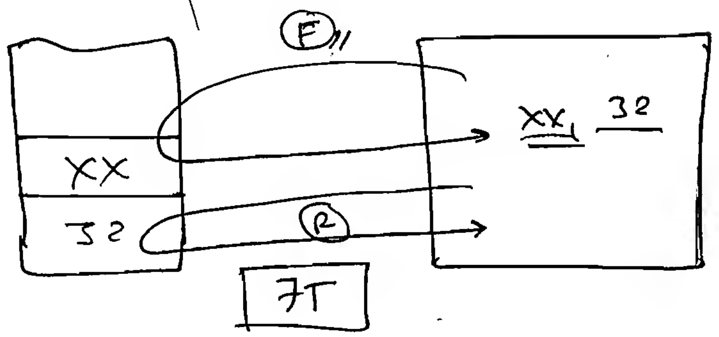
e.g MOV C, A



⇒ ② 2- Byte Instruction:

⇒ MVI C, 32 ⇒

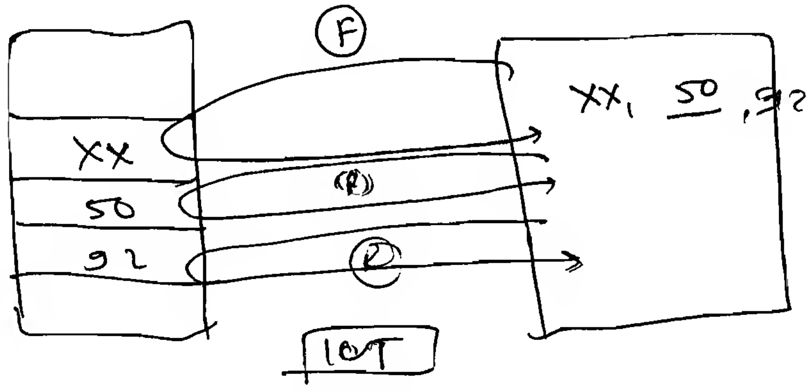
Opcode = XX



③ 3- Byte Instruction:

e.g. LDA 9250

Opcode: XY



→ In Any instruction cycle the first machine cycle is opcode fetch machine cycle.

Eg.

① XRI 46_H ⇒ 2B

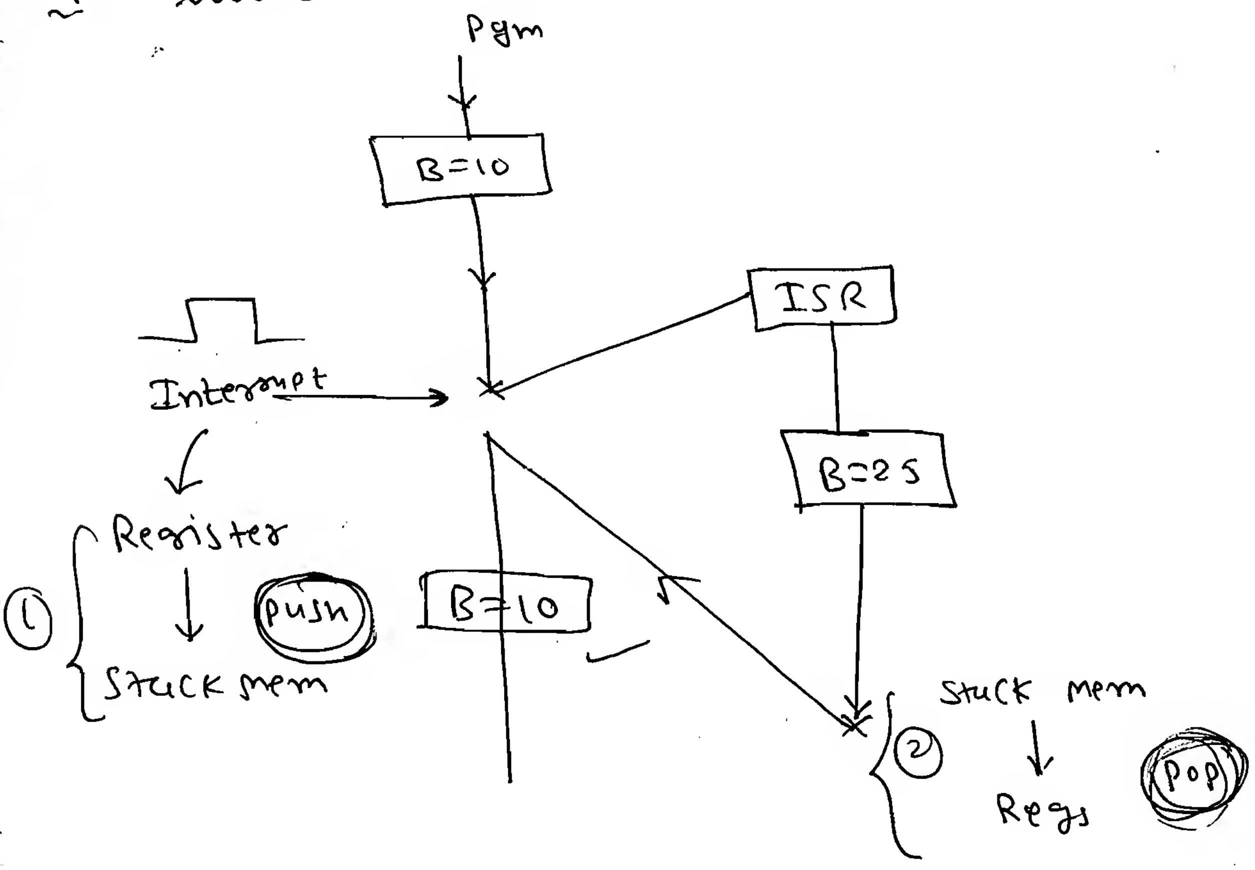
② Push PSW ⇒ 1B.

③ LDAX B ⇒ 1B.

④ LHLD 8300_H ⇒ 3B.

⑤ DAD SP ⇒ 1B.

* Stack:



⇒ SP_{16} = Stack Pointer$_{16}$:

∴ $\boxed{\text{LIFO = Last in First out.}}$

① $\boxed{\text{Push RP}}$

- Push B
- Push D
- Push H
- Push PSW

Predecrement

=> decrement SP + push RP_H

=> decrement SP + push RP_L

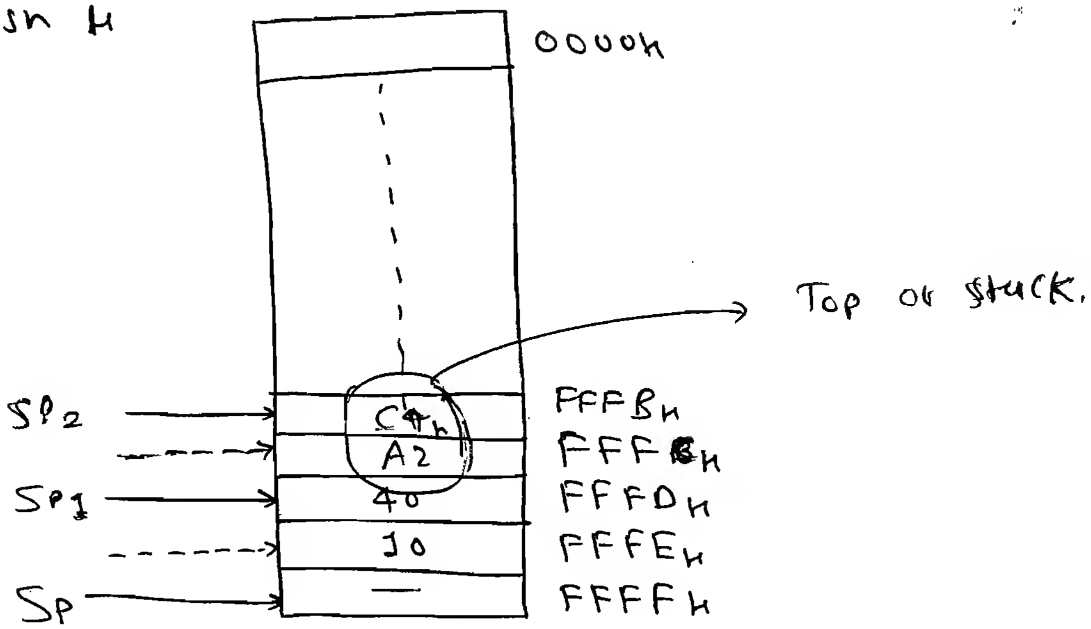
$RP_{16} = RP_H + RP_L$

Eg. Given SP = $FFFF_H$; BC = 1040_H ; HL = $A2C4_H$.

① push B ② Push H.

Ans: ① push B.

② push H

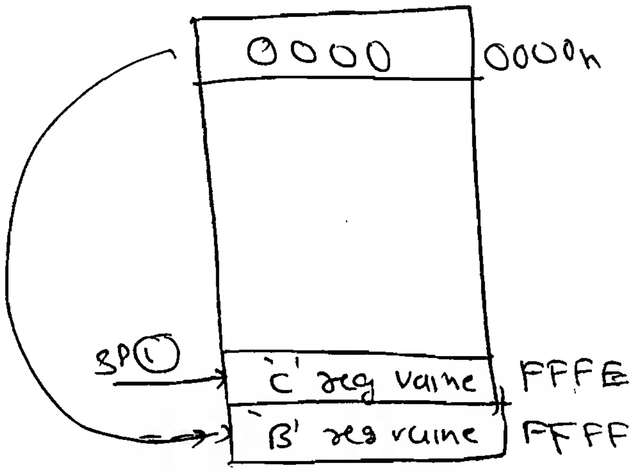


P1) SP= 0000H.

① PUSH B.

B, C, regs ⟶ mem Locations = ?

$$\begin{array}{r} 0000 \\ -\quad 1 \\ \hline FFFF \\ -\quad 1 \\ \hline FFFE \end{array}$$



② $\boxed{\text{Pop rp}}$ → $\boxed{\text{Post increment}}$

PoB B

Pop D ⇒ get 1 Byte from Stack into rp_L + Increment SP.

PoP H

Pop PSW ⇒ get 1 Byte from Stack into rp_H + Increment SP.

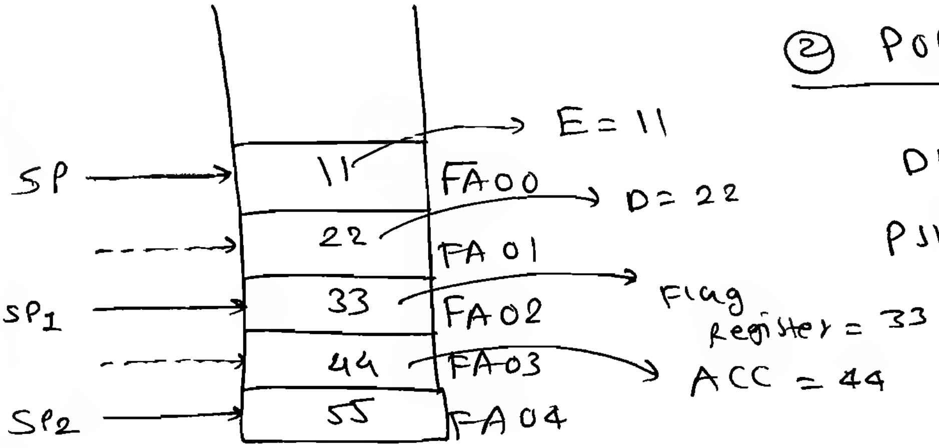
Eg. Given

① PoP D

② PoP PSW

DE = ?

PSW = ?



DE = 2211_H

PSW = 4433_H.

P2) let SP = FFFF ; BC = 5065_H ; HL = $A1F4_H$

① PUSH B

② PUSH H

③ PoP D

④ PoP PSW

→ (LIFO)

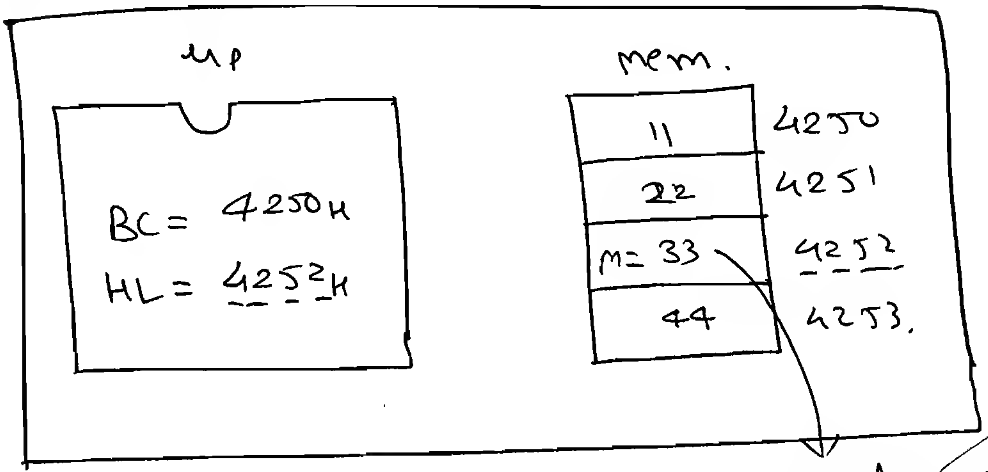
HL → DE ⇒ DE = $A1F4_H$.

BC → PSW ⇒ PSW = 5065_H.

* Types of Instructions:

1) Arithmatic
2) Logical
3) Data Transfer
4) Branching
5) Machine related , I/O.
6) Additional.

* Reference:



mov C, M

M → C.

present memory locations.

$(4250) = 11_H.$

$(4253) = 44_H$

$((BC))$

⇓

$(4250) = 11_H.$

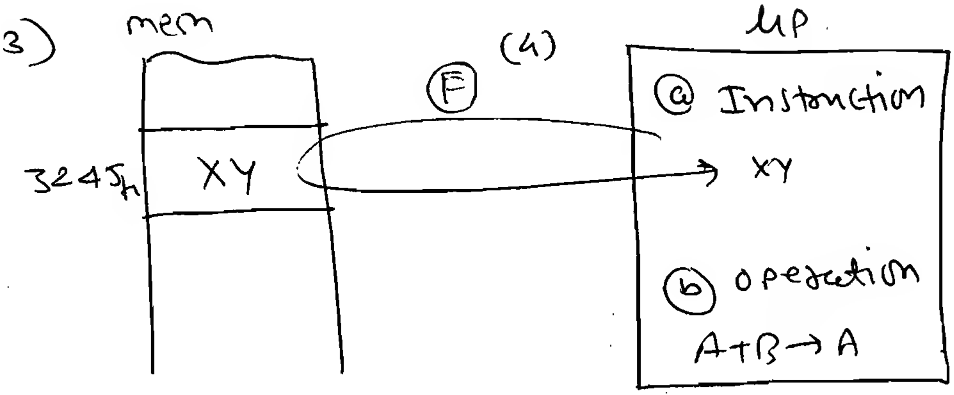
$M = ((HL)) = 33_H.$

① Arithmatic Instruction:

Sr. No.	Instruction	operation.	B|M|T	Flags affected
1)	$\boxed{\text{ADD R}}$	A+R ⟶ A	1|1|4 (F)	All

E.g. 1) ADD B ⇒ (2) opcode = XY

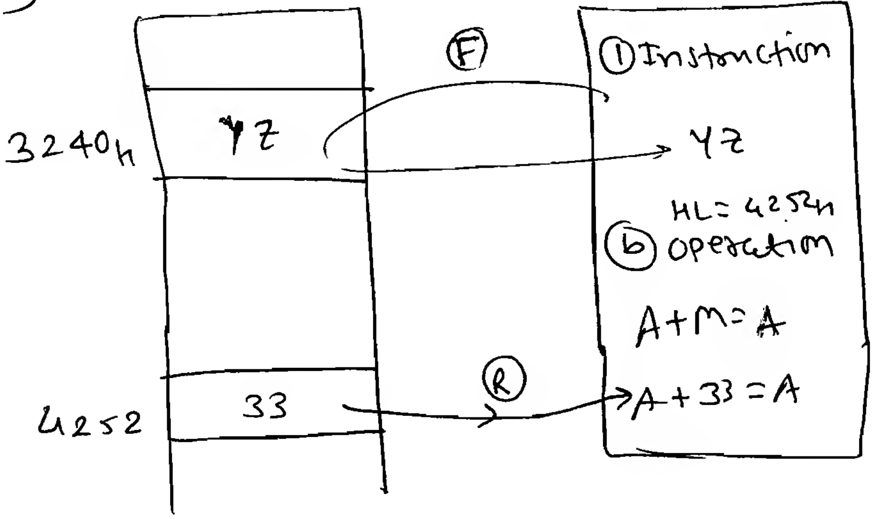
3)



2)	$\boxed{\text{ADD M}}$	A+M ⟶ A	1|2|7|(F.R)	All.

E.g 1) ADD M ⇒ (2) opcode = XY.

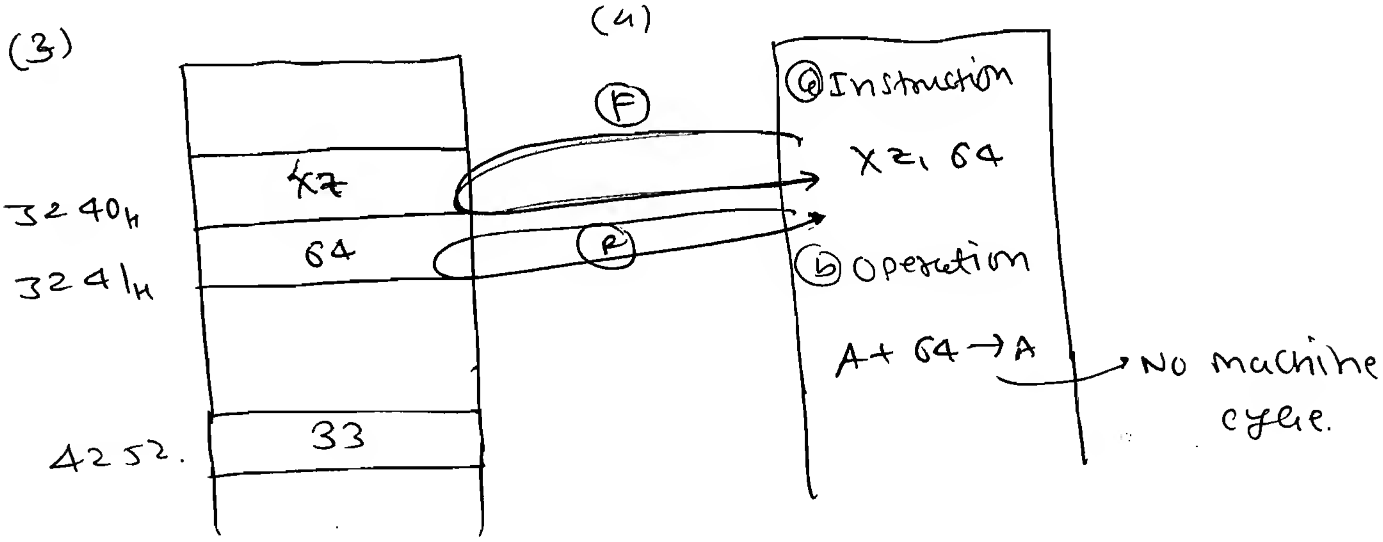
3)



3) ADI 8 bit data A+8bitdata→A 2|2|7 (F.R) All

Eg. (1) ADI 64_H ⇒ (2) Opcode = XZ, 64_H.

(3) (4)



2) (1) SUB R A-R → A 1|2|7 (F.R.) All.

(2) SUB M A-M → A

(3) SUI 8 bitdata A-8bitdata → A

3) (1) ADC R A+R+CY → A 1|1|4 (F) All.

ADC M A+M+CY → A 1|2|7 (F.R.).

ACI 8bit data A+8bitdata+CY→A 2|2|7 (F.R.)

4) SBB R A - R -CY → A 1|2|7 (F.R.) All.

SBB M A -M -CY → A

SBI 8bit data A-8bit data -CY → A

5) INR R R+1 → R 1|1|4 (F)

$\boxed{\text{INR M}}$ M+1 → M 1|3|10T (F, RW).

DCR R R-1 → R 1|1|4.

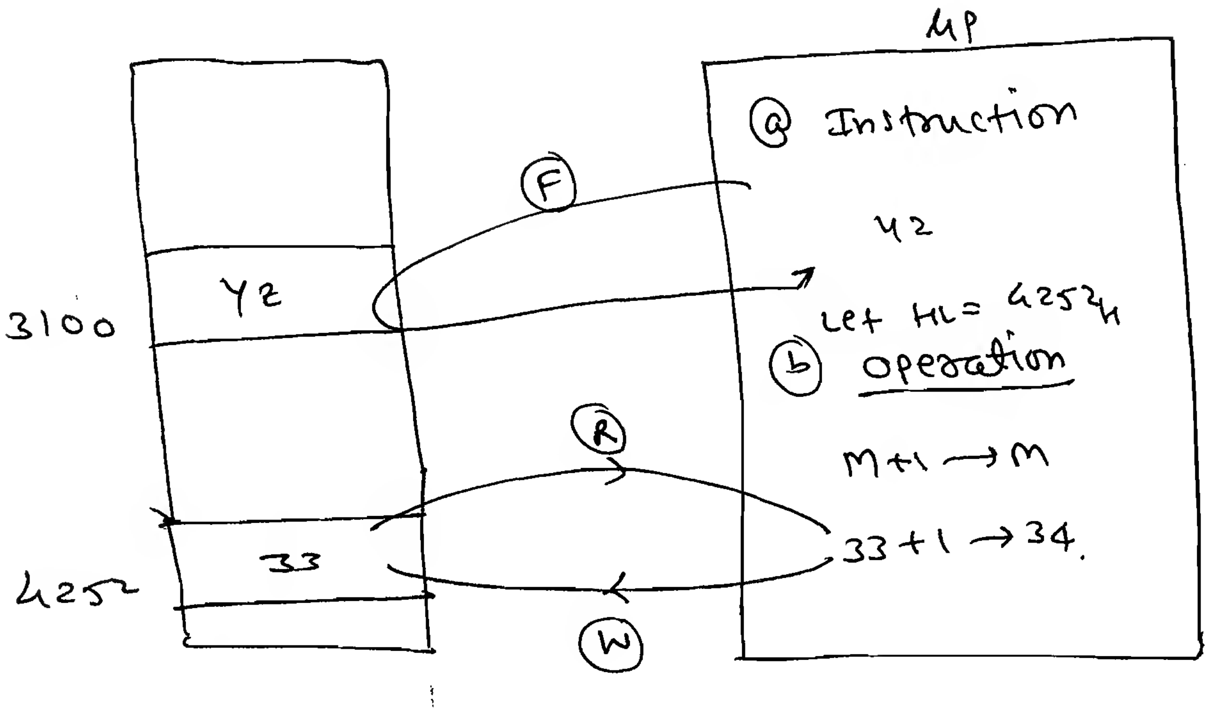
$\boxed{\text{DCR M}}$ M-1 → M 1|3|10.

S, Z, P, AC affected. but CY not affected.

1) INR M   2) opcode = YZ.

3) Memory.

4)



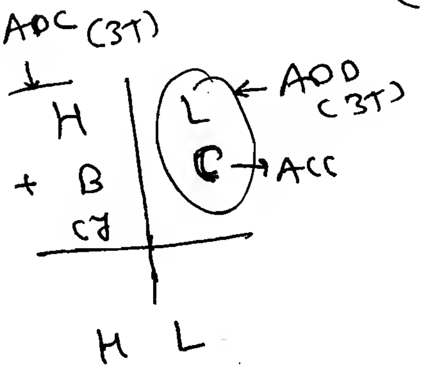
6) INX Rp   Rp+1 → Rp   1|1|6(S) ✓   No flags are affected [∵ operation doesn't take place in ALU].

DCX Rp   Rp - 1 → Rp   1|1|6(S) ✓

INX B
INX D
INX H
INX SP

7) DAD Rp   HL+Rp → HL   1|3|10 (FBB) ✓   * only carry flag if carry occurs out of 16 bits.

DAD B
DAD D
DAD H
DAD SP



② Logical operations:

Sr. No.		Instruction	operation	B/M/T	Flags affected.
1).	a)	ORA R	$A \vee R \rightarrow A$	1/1/4	* $\boxed{CY=0}$ S, Z, P are afferted.
		ORA M $\rightarrow$ ①	$A \vee M \rightarrow A$	1/2/7 (FR)	
		ORI 8bit data ① $\rightarrow$ F ② $\rightarrow$ R	$A \vee$ 8bit data $\rightarrow A$	2/2/7 (F.R).	
	b)	XRA R	$A \oplus R \rightarrow A$	1/1/4	
		XRA M	$A \oplus M \rightarrow A$	1/2/7 (FR)	
		XRI 8 bit data	$A \oplus$ 8bit data $\rightarrow A$	2/2/7 (F.R.)	
	c)	ANA R	$A \wedge R \rightarrow A$	1/1/4	
		ANA M	$A \wedge M \rightarrow A$	1/2/7 (FR)	
		ANI 8bit data	$A \wedge$ 8bit data $\rightarrow A$	2/2/7 (FR)	

2) ~~CMP~~.

$\boxed{\text{CMP R}}$ $\quad A - R \quad$ 1/1/4

- $A < R \Rightarrow CY=1, Z=0$
- $A = R \Rightarrow CY=0, Z=1$
- $A > R \Rightarrow CY=0, Z=0.$

* Register values are unchanged. only flags are affected.

S, P, AC are affected.

CMP M $\quad A - M \quad$ 1/2/7

- $A < M \Rightarrow CY=1, Z=0$
- $A = M \Rightarrow CY=0, Z=1.$
- $A > M \Rightarrow CY=0, Z=0.$

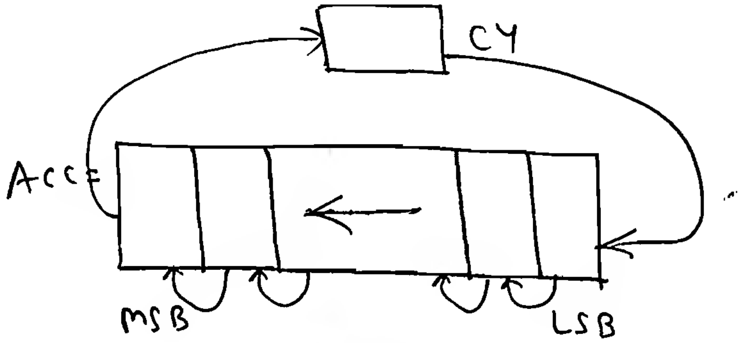
CPI 8 bit data   A - 8bit data   1|1|7

A < 8-bit data => CY=1, Z=0

A = 8·bit data => CY=0, Z=1

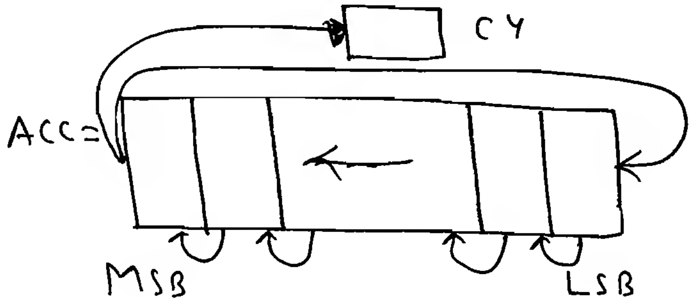
A > 8 bit data => CY=0, Z=0.

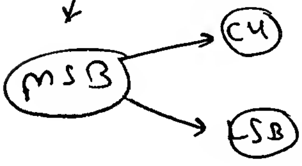
3) RAL (with CY)



only carry flag is affected.

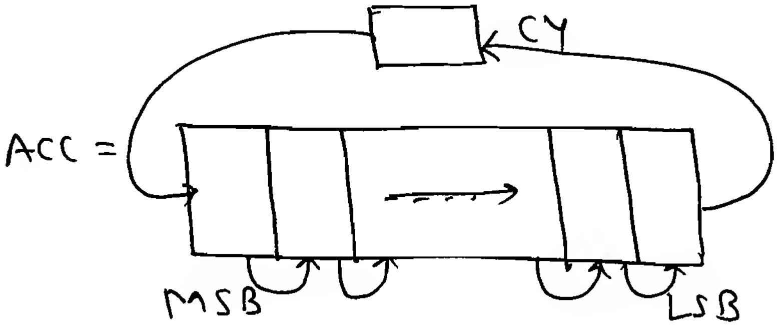
RLC (without carry).





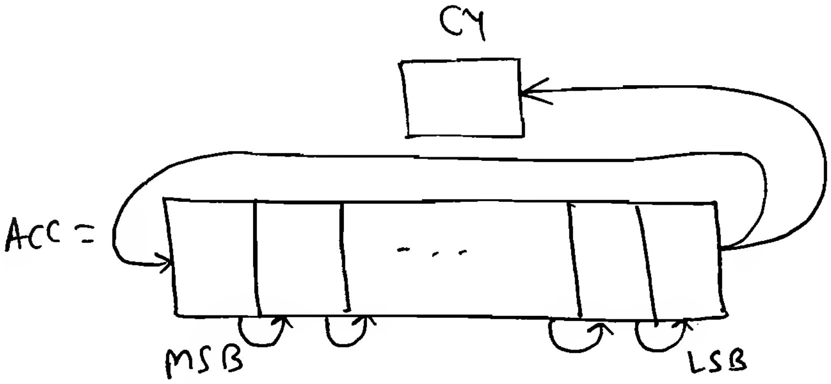
1|1|4 ✓

4) RAR with carry



only 'CY' flag is affected.

RRC without carry.



1|1|4

4) CMA $\bar{A} \rightarrow A$ 1|1|4 No flags are affected (∵ operation is within Acc.).

100

✓ CMC $\overline{CY} \rightarrow CY$ 1|1|4

✓ STC CY = 1

} → only 'CY' flag.

Ex-1

let, $A = F2_H$.

CY = 1.

$\boxed{XRA\ A} \Rightarrow A \oplus A \longrightarrow A$.

$F2_H \oplus F_2 \longrightarrow 00_H$.

Ans: $A = 00_H$

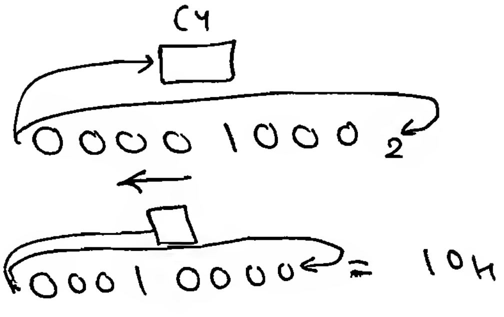
CY = 0.

Ex-2 Let $A = 08_H$

1) RLC .

3) RLC

Ans:



$A = 0000\ 1000_2$

$= 0001\ 0000 = 10_H$

$A = 10_H = 16_{10}$.

A = 00100000 = 20_H.

All flags are cleared.

Ex-3 Let, $A = 3C_H$;

CPI 7F.

A = ?

S = ? CY = ?

Z = ? AC = ?

P = ?

Ans: CPI $7F_H$.

$\therefore \rightarrow A - 7F.$

$= 3C_H - 7F_H.$

$= +bd_H.$

$$\begin{array}{r} \overset{1\,2}{\overset{2}{}}\;1C \\ 3C \\ -\;7F \\ \hline B\,D_H \end{array}$$

$= 10111101$

$\therefore A = 3C_H.$

$S = 1$

$Z = 0$

$P = 1$

$CY = 1$

$AC = 1.$

Ex-4

→ In a µP XOR (P,Q) is defined as $P \oplus Q \Rightarrow P$

what is the f^n of the following program.

XOR (r_2, r_1). → $r_2 = r_2 \oplus r_1$.

XOR (r_1, r_2) → $r_1 = r_1 \oplus r_2$. $\Rightarrow r_1 = r_1 \oplus r_2 \oplus r_1 = 0 \oplus r_2 = r_2$ $\boxed{r_1 = r_2}$

XOR (r_2, r_1) → $r_2 = r_2 \oplus r_1$.

$\Downarrow$

$r_2 = r_2 \oplus r_1 \oplus r_2.$

$r_2 = 0 \oplus r_1$

$\boxed{r_2 = r_1}$

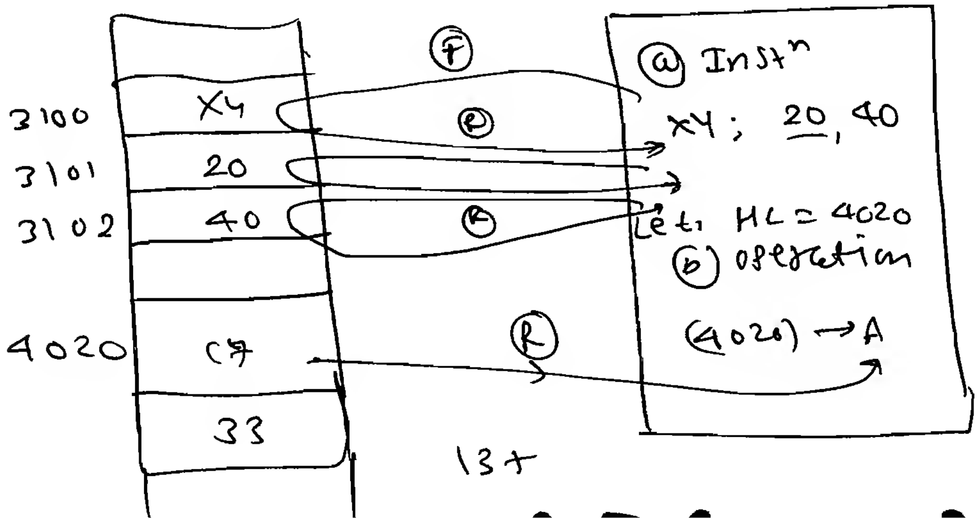
Ans: $\boxed{\text{Swap. } r_2 \longleftrightarrow r_1}$

③ Data transfer Instruction:

Sr. No.	Instruction	Operation	B|M|T.
1)	Mov Rd, Rs	$R_s \xrightarrow{copies} R_d$	1|1|4.
	Mov R, M	M → R	1|2|7 (FR)
	Mov M, R	R → M	1|2|7 (FW).
	MVI R, 8bit data	8 bit data → R	2|2|7 (FR)
	MVI M, 8 bit data ① ②	8 bit data → M ③	2|3|10 (FRW).
	LXI Rp, 16 bit data ① ③ ② μp	16 bit data → Rp	3|3|10 (FRR).
2)	LDA 16 bit address ① ③ ②	(16 bit addr) → A ④	3|4|13. (F.R.RR).

1) LDA 4020   2) opcode = X4, 20, 40.

3) Memory   4) μp



	Adress
M1	PC = 3100H
M2	PC = 3101h.
M3	PC = 3102h
M4	PC = 4020H

2) STA 16 bit address $A \rightarrow$ (16 bit address) ✓

↳ ④ write

$\boxed{3 | 4 | 13.\ (F.R.R.W)}$

3) LDAX Rp $((Rp)) \rightarrow A.$ ✓

1 | 2 | 7T (F.R.).

e.g. LDAX B

$((BC)) \rightarrow A.$

$(4250) \rightarrow A.$

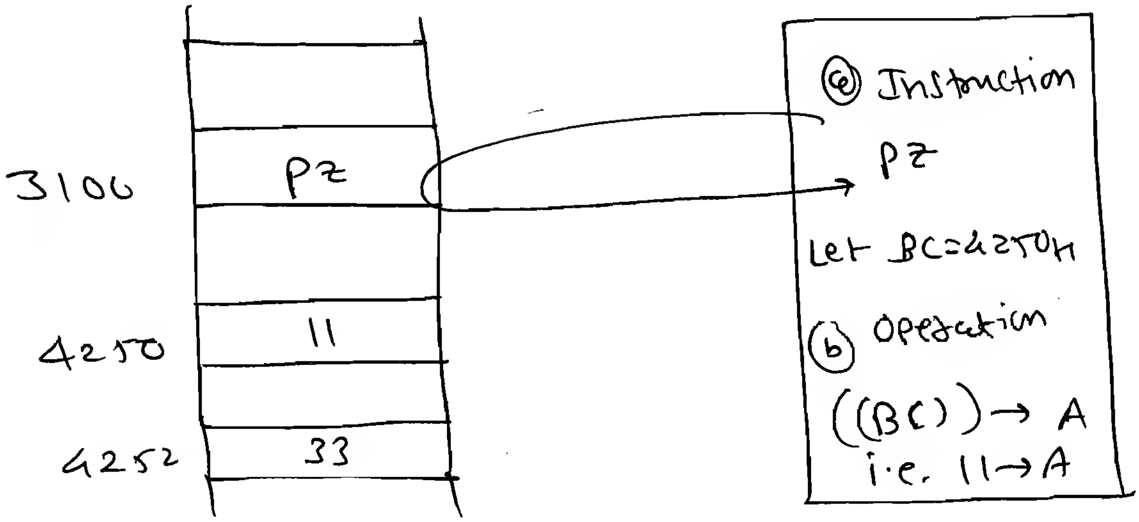
$\boxed{11 \rightarrow A}$

1) LDAX B

2) Opcode = PZ.

3) Memory

4) μP



ⓐ Instruction

PZ

Let BC=4250H

ⓑ Operation

$((BC)) \rightarrow A$

i.e. $11 \rightarrow A$

→ LDAX B.

~~LDAX D.~~

~~LDAX H~~ ⇒ $((HL)) \longrightarrow A.$

⇒ i.e. $M \longrightarrow A$ = MOV A, M.

⇒ STAX Rp $A \longrightarrow ((Rp))$ 1 | 2 | 7T (FW). ✓

④ LHLD 16 bit address ~~(16 bit addr)~~ → (16 bit addr) → L ④

① ③ ② ④

(16 bit addr +1) → H ⑤

3/5/16 (FRR RR)

e.g.

LHLD 4251

(4251) → L i.e. L = 22.

(4251+1 = 4252) → H i.e. H = 33.

SHLD 16 bit address ✓

L → (16 bit address)

H → (16 bit address +1)

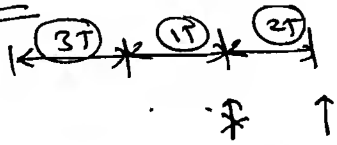
3/5/16 (FRR WW).

⑤ XCHG HL → DE 1/1/4.

⑥ a) PCHL HL → PC ✓

b) SPHL HL → SP ✓

Eg: 2 HL → PC

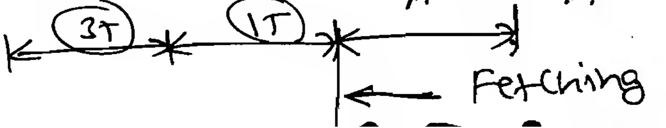


* FEO is not possible.

NOTE:→

→ In 8085 μP opcode fetch machine cycle is extended by two T-states if the PC (or) SP value is changing. This results in opcode fetch machine cycle as shown in the following example.

Eg: 1 ADD B ⇒ 4T

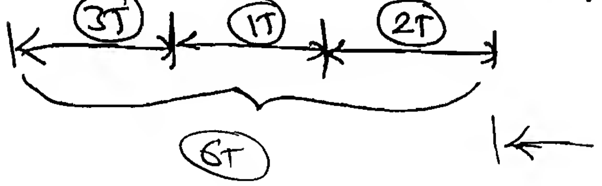


* $\boxed{\text{INX Rp}}$

↓

$\boxed{\text{INX Sp}}$

Sp+1 → Sp

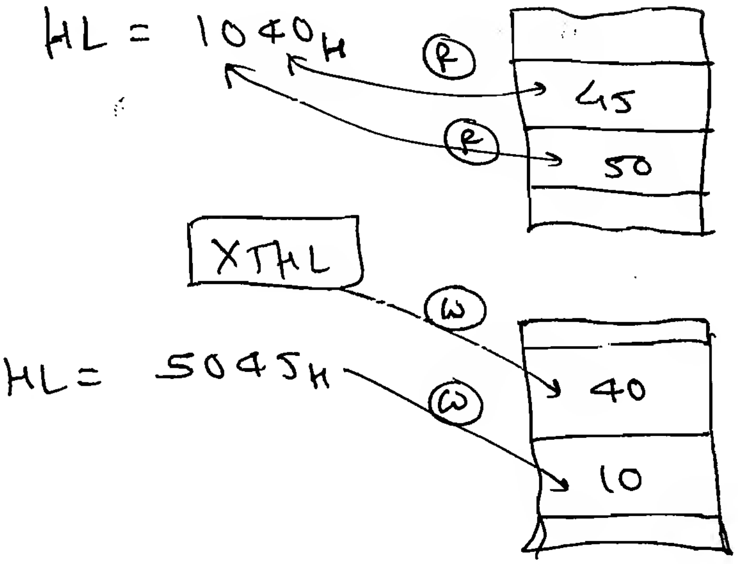


6T

7) XTHL   Top of the Stack ↔ HL   1|5|16

F → 1

Eg.:->



8) a) $\boxed{\text{Push Rp.}}$ 1, F

Pre-decrement

$Rp_H \rightarrow ((Sp-1))$. (2) W

$Rp_L \rightarrow ((Sp-2))$. (3) W

and $\boxed{Sp = Sp-2.}$

1|3|12

(S W W
6 3 3)

b) $\boxed{\text{Pop Rp}}$ 1, F

post-increment

$((Sp)) \rightarrow Rp_L$. (1)

$((Sp+1)) \rightarrow Rp_H$. (R)

and $\boxed{Sp \rightarrow Sp+2}$

1|3|10

(F R R
4 3 3)

Eg: Let, $(F250)_H = 4B_H$.

$(F251)_H = 50_H$

$(F252)_H = F2_H$

$(FF53)_H = 04_H$.

① LXI, H, F250. → HL = F250 i.e $M = 4B_H$.

INX H. → HL= F251 i.e. ~~[illegible]~~ $M = 50_H$.

DCR M → $M = 4F_H$.

MOV C, M. → $C \rightarrow 4F_H$. ⇒ $M \xrightarrow{\text{Copied}} C$

C = ?

∴ $\boxed{C = 4F_H.}$

② LHLD F251. → $L = 50_H$, $H = F2_H$. HL = F250h

(F251) → L i.e. $\boxed{L = 50}$ $H = F2_H$

(F252) → H i.e

i.e. $M = 4B_H$.

INX H → $H = F2_H$, $L = 51_H$. ~~F2~~ → i.e. $M = 50_H$.

INR M → $M_{+1} = 51_H$.

MOV C, M. → $C = 51_H$.

C = ?

$\boxed{C = 51_H.}$

③ LXI D, F252. → DE= $F2\cancel{5}2_H$

LXI H, F253. → $\boxed{HL = F253_H.}$ ⇒ $M = 04_H$.

LDAX D. ((DE)) → A i.e. (F252) → A. A → F2.

ADD M $A + M = F2 + 04_H = A$.

A = ? $A = C6_H$ with CY=1.

Ex-4 Find the Value of Accumulator after executing the following programm.

④	③ Address	① Mnemonics	② Opcode
↓ execute	3000, 01	MVI A, 36	3E, 36 => A = 36
	3002, 03	ADI 40H	82, 40. => A + 40 => A = 76
	3004, 05, 06	STA 3007	32, 07, 30.
	3007	~~XRA A~~ HLT	~~AF~~ 76
	3008	HLT	76.

$A = 76_H.$

* 'XRA A' instn is not executed because its opcode 'AF' is Replace by '76' which is opcode of HLT Instruction.

(4) Branching Instructions:

* Unconditional Jump.

Sr. No.	Instruction	operation	B/M/T	flag.
1) (a)	JMP 16 bit address ↳① ↳③ ↳②	16 bit addr → PC	3/3/10.	(FRR).

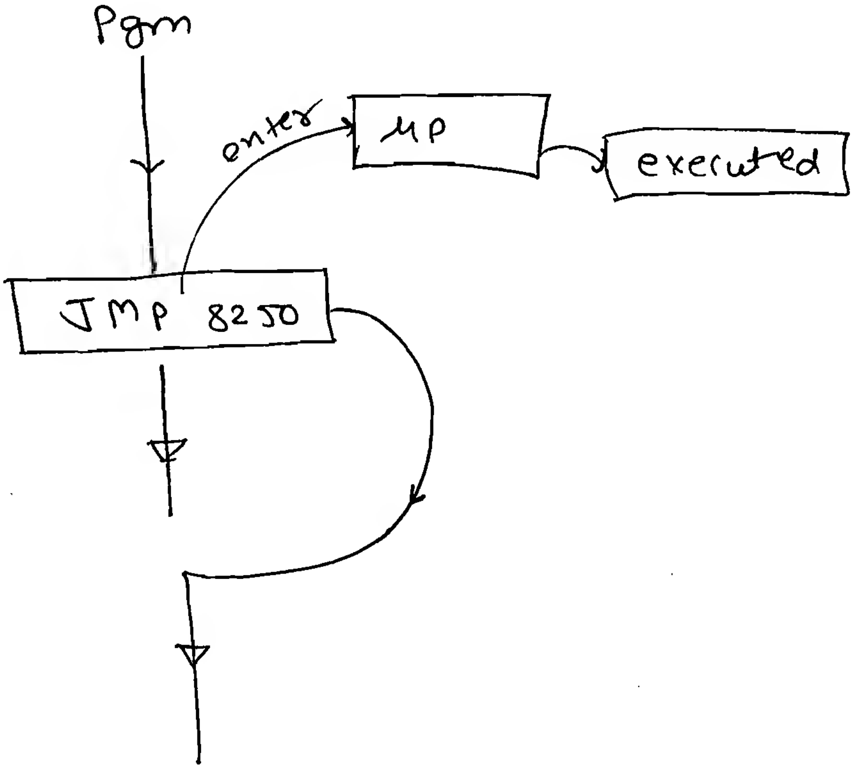
e.g. PC => 3800

3806,07,08

PC => ~~3807~~

PC = 8250

8257



* Conditional Jump.

(b) JC 16 bit address => If $\boxed{CY=1}$ ← True

↳① ↳② and CY=?

16 bit addr → PC

3/ $\frac{2}{3}$ / $\frac{7}{10}$ ← False ← True

True

→ JNC 16 bit address => If $\boxed{CY=0}$

– JZ 16 bit add. => If $\boxed{Z=1}$

→ JNZ 16 bit add. => If $\boxed{Z=0}$

→ JM 16 bit address ⇒ True

Jump on minus

If $\boxed{S=1}$

→ JP 16 bit address ⇒ If $\boxed{S=0}$

Jump on plus

→ JPE 16 bit address ⇒ If $\boxed{P=1}$

Jump on parity even.

JPO 16 bit address ⇒ If $\boxed{P=0}$

Jump on parity odd.

* Unconditional Call:

2) (c) CALL 16 bit address.

5 → ①, R ③, R ②

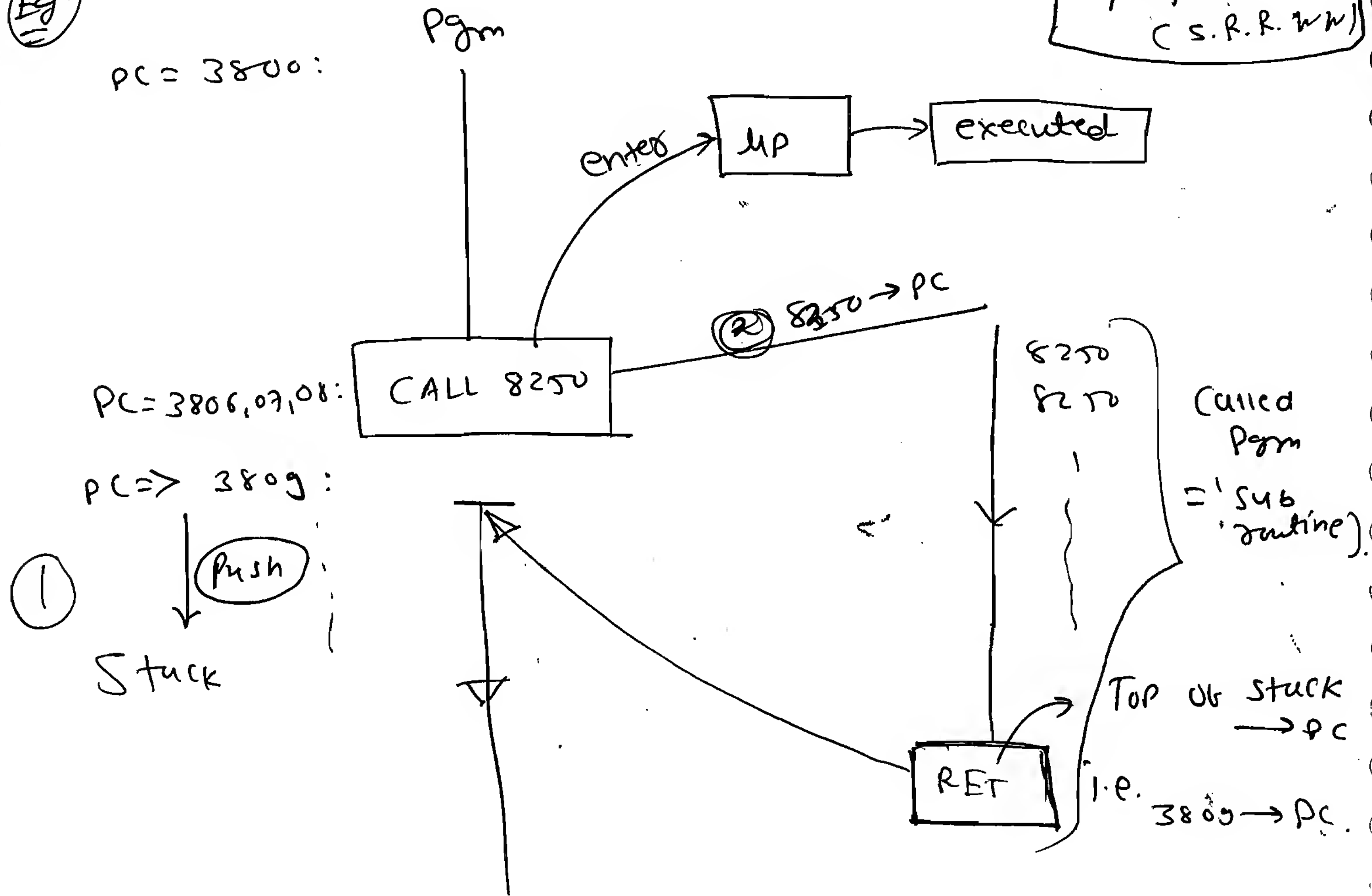
(a) $PC_H \xrightarrow{②\,Ⓦ} ((Sp-1))$

$PC_L \xrightarrow{③\,Ⓦ} ((Sp-2))$ and $sp = sp-2$.

(b) 16 bit addr → PC.

3/5/18 (S.R.R. [illegible])

Eg.



* Conditional CALL:

(b) CC 16 bit add. ⟹ If $\boxed{CY=1}$

S (1) (3) (2)

(a) $PC_H \xrightarrow[(4)]{} ((SP-1))$

~~(4)~~ $\rightarrow PC_L \xrightarrow[(5)]{} ((SP-2))$

(b) 16 bit add. → PC.

(3 | 5 | 18
S.R.R.WW)

SP = SP - 2. ✓

True

→ CNC 16 bit addr. (a) if $\boxed{Cy=0}$

3 | 2 | 9 ← False

3 | 5 | 18 ← True ✓

→ CZ 16 bit add. if $\boxed{Z=1 \quad \cancel{Z\neq 0}}$

→ CNZ 16 bit add. if $\boxed{Z=0.}$

⟹ CM 16 bit add. ⟹ If $\boxed{S=1.}$

CP 16 bit add ⟹ If $\boxed{S=0.}$

CPE 16 bit add ⟹ If $\boxed{P=1.}$

CPO 16 bit add ⟹ If $\boxed{P=0.}$

3) a) RET ⟹ Top of stack → PC

(F) (1)

similar to pop Instⁿ.

i.e. $((SP))$ (R) (2) ⟶ PC_L

$((SP+1))$ (R) (3) ⟶ PC_H.

and SP = SP+2.

✓ $\boxed{1 | 3 | 10 \quad (F.R.R.)}$

b) Conditional

RC ✓
RNC ✓
RZ ✓
RNZ ✓
RM ✓
RP ✓
RPE ✓
RPO. ✓

1 | 1/3 | 4+2 = 6T ← False

10 ← True

Ex-(1)

8204 : CALL 8207 → ② PC = 8207

8207 : POP H ⇒ Top of Stack → HL

i.e 8207 → HL

HL = ?

① ↓ Stack

$\boxed{HL = 8207_H}$

Ex - 2 In the following program determine the return address of the Subroutine

LXI D, 705F ⇒ DE = 705F

8205 : CALL sub1.
↓
PC = 8208

8208 :

① push stack

sub1 : POP H HL = 8208

DAD D HL + DE = HL = FF67

PUSH H ⇒ TOP of the stack = FF67

RET.
⇓
PC ⇒ TOP of the stack

PC = FF67h

PC ⇒ FF67 :

Ex-3 Determine the acc. value after executing the following program

Main pgm

CALL sub1

① ↓ stack XX YY

Subroutine

Sub1 : XRA A ⇒ A = 00h

CALL 4260H.

4260 : INR A ⇒ A = 01h | A = 02H.

① push-stack RET ⇒ PC = 4260H i.e XXYY → PC.

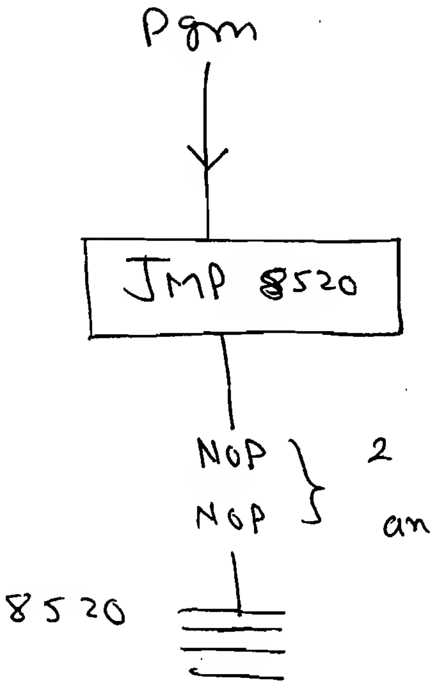
A = ? = ~~0~~. 02h

(5) Machine related, I/O Instruction: ///

1) a) NOP   No operation   1/1/4. (a) Delay (b) To avoid Addr. Relocation.

b) HLT   μp is Halted   1/2 (OR) more / 5 (OR) more.

E.g. NOP



2 Byte Instn is deleted and replaced with NOP Instn.

* HLT :

→ After executing the Halt instruction:

(i) it enters into Halt acknoledgement machine cycle.

(ii) the Buses are tristated.

(iii) Registers values are unaffected.

(iv) Either reset (OR) interrupt is required to bring the μp out of the halt state.

⇒ As a Good program we have to use BPI (Break point ~~[illegible]~~ Interrupt) instead of HLT* as it takes less time as that of HLT for executing.

2)

Used in "I/O mapped I/O mode".

ⓐ In 8 bit address. [8 bit addr.] → A   2 | 3 | 10 (FRI) (4+3+3)

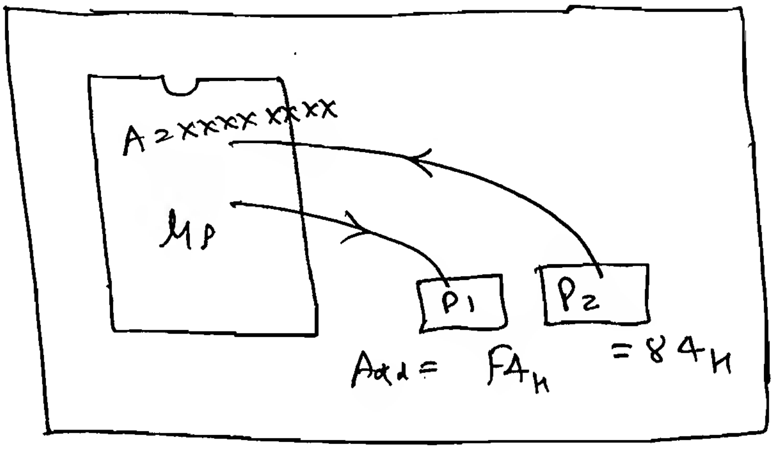
① ② Input device ③

ⓑ OUT 8 bit add. A → [8 bit add]   2 | 3 | 10 (FRO)

Ⓕ ① Ⓡ ② Output device Ⓦ ③

* IN 84_H

OUT $F4_H$.



Ex-1 : After executing the following program determine which port receive^s the data

```
        MVI A, 87H.  => A = 87H
        MOV B, A.    => B = 87H          Ans:- PORT 2.
Again:  JMP   Next
        XRA   B
        OUT   PORT 1
        HLT                                        0000 0000
                                                 + 1000 0111
                                                   1000 0111
Next:   XRA ~~B~~ B ~~B=>00H~~ A⊕B => A = 00H. | 00H ⊕ 87H. => 87H.
        JP    Again                        1000 0111
        OUT   PORT 2 <- Acc = 87H           +ve  False
        HLT
```

=> 87_H is sent to PORT2.

Ex-2 The following Programm reads signed no. in 2's Comp. form from port -1 when the execution of the program stops?

```
Again : In port 1        => PORT 1 → A = Sxxx xxxx_2
        RLC              => CY=S
        JNC Again.
        RRC  If CY=1 , S=1  -ve no.
        HLT
```

if CY=0, S=0 +ve no.

=> the above program stops the execution when it reads a -ve no. from port+1.

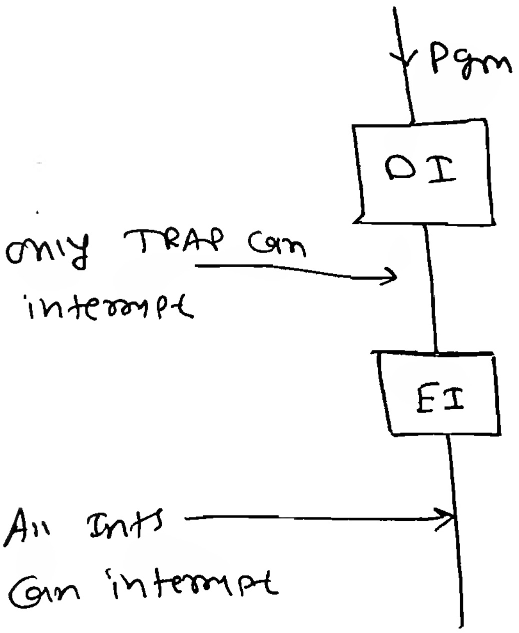
⑥ Additional Instruction:

① ⓐ DI   Disable Interrupts   1/1/4

=> { RST 7.5, 6.5, 5.5, INTR are disabled.

ⓑ EI   Enable Interrupts   1/1/4

=> All Ints are Enabled.



2) DAA Decimal adjust Acc 1/1/4 ⇒ All flags are affected.

(BCD addition).

27_{10}
$+ 36_{10}$
63_{10}

```
MVI  A, 27.
MVI  B, 36
ADD  B  => A+B → A = 5DH
DAA     => A = 63
HLT
```

```
 27  =   0010 0111
+36  = + 0011 0110
         0101 1101   x
       + 0000 0110
         0110 0011
          6    3
       = 63
```

Ex-1 Determine the PC Value after executing the following Programm.

```
LXI  H, 8A79  => HL = 8A79.
MOV  A, H     => A → 8A
ADD  L   →  A+L → A → 8A+79.
DAA         A → 69.
MOV  H, A   H → 69.
PCHL        PC → 6979.
PC = ?
```

```
   1000 1010
 + 0111 1001
 1 0000 0011
   0110 0110
   0110 1001
     69.
```

NOTE: → DAA Instruction is effective only after add instn it doesn't work for BCD subtraction.

Ex-2 Determine the fn of the following Program in B & C Register Contain BCD numbers.

```
MVI   A, 99.   =>  A = 99.
SUB   C  ->  A -> 99 - C =   9's Comp.(C)
INR   A  ->  A+1 -> A ->  10's Comp. C C).
ADD   B  ->  A + 10's C -> A.
DAA      ->  A -> B-C.   A ->   B-C.
HLT      ->
```

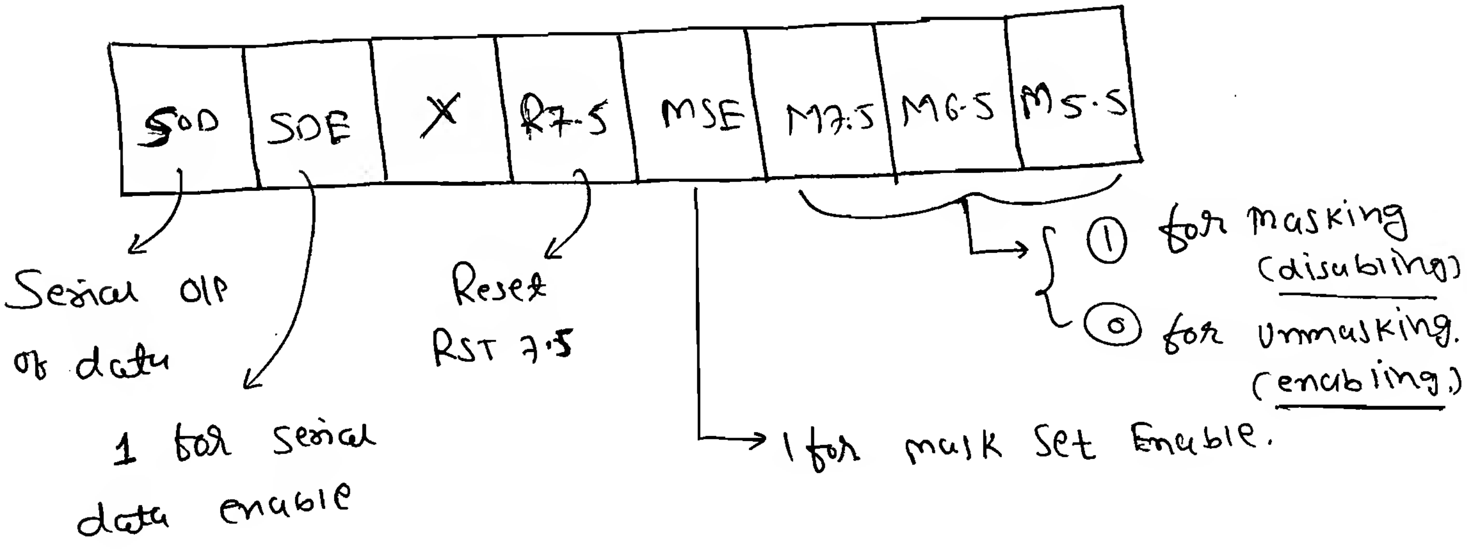
Let, B = 64; $99 - C = 99 - 42 = 57 \rightarrow A$

C = 42; 10's of A' ⇒ A = 58H.

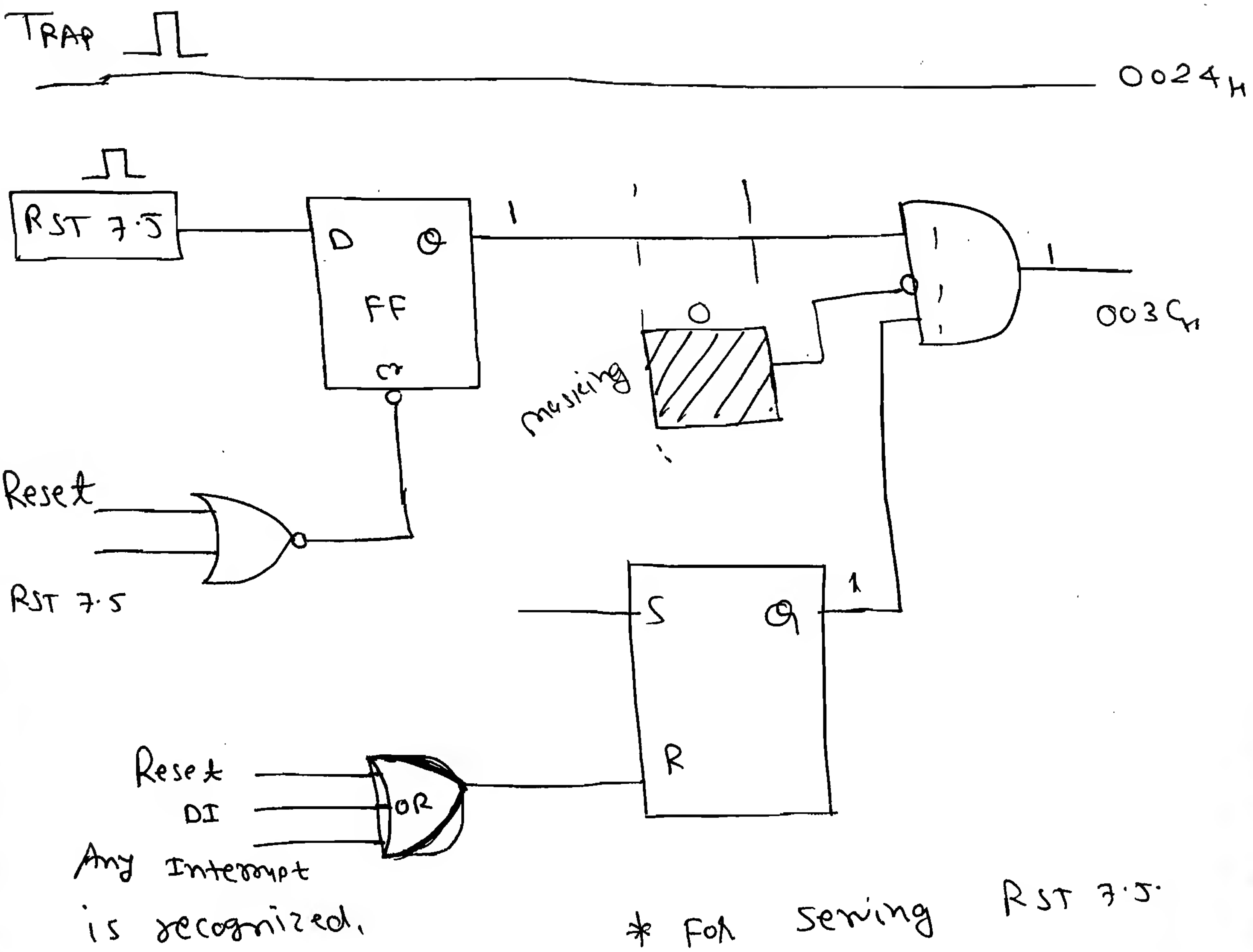
3) a) SIM (set Interrupt mask):

i) selective disabling Interrupts.

ii) serial output of data.



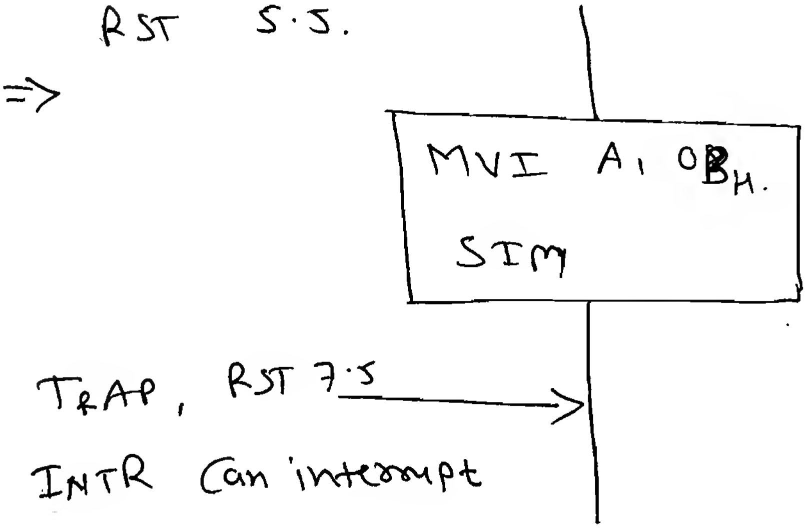
1	1	4



* For serving RST 7.5 the required values are Q=1, masking = 0, Q=1.

Ex-1: Write instructions to mask RST 6.5 and RST 5.5.

=>



```
MVI A, 0BH.
SIM
```

=> RST 6.5, 5.5 are masked (disable).

Ex-2 Determine the O/P of the following program. 117

```
Start: MVI A, 40H.      => A = 0100 0000
       SIM                 '0' is sent on SOD pin of µP.
       Call 10msDelay
       MVI A, C0H.      => A = 1100 0000₂
       SIM                 '1' is sent on SOD in pin of µP.
       Call 10msDelay
       Jmp start.
```

b) RIM (Read Interrupt Mask)

(a) To know status of Interrupts.
(b) To read serial input [on SID pin of µP].

1/1/4

Acc=>

SID	I7	I6	I5	IE	I7.5	I6.5	I5.5

SID → Serial input data

I7, I6, I5 → If 1 => Int is pending

IE → 1 → Ints enabled. 0 → Ints disabled.

I7.5, I6.5, I5.5 → 1 = Int is masked (disabled) 0 = Int. is Unmasked (enabled).

Ex-1 After executing RIM Instruction the accumulator value is BC_H. Determine the status of the interrupt and the serial input.

Ans: $A = BC_H = 1\,011\,1\,100_2$

1 → serial bit on SID pin = '1'

011 → RST 6.5, 5.5 are pending [∵ TRAP Int ISR is getting executed)

1 → Ints are enabled

100 → RST 7.5 is masked.

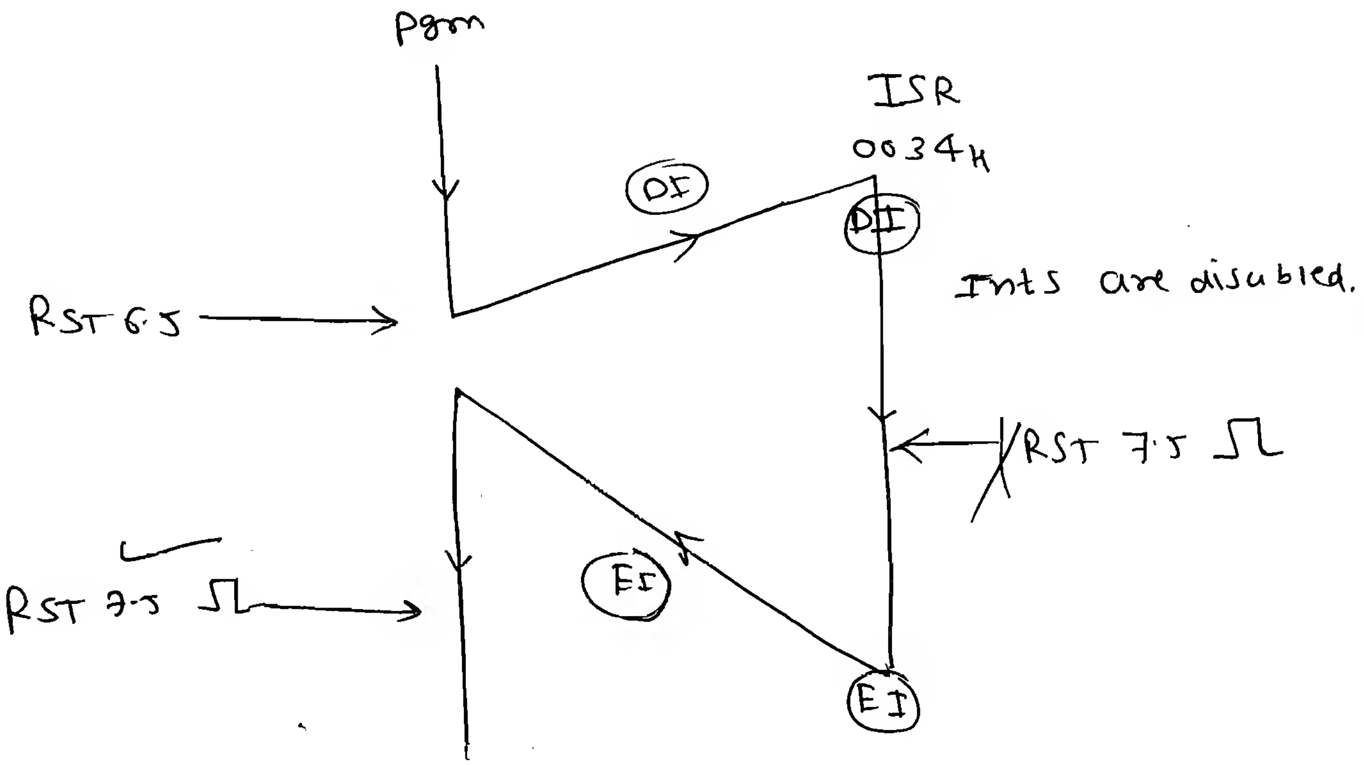
⇒ In 8085 µP the interrupts are disable in the following cases:

(i) when 'DI' instⁿ is executed.

(ii) when µp is Reset.

(iii) when µP Recognizes any one interrupt

⇒



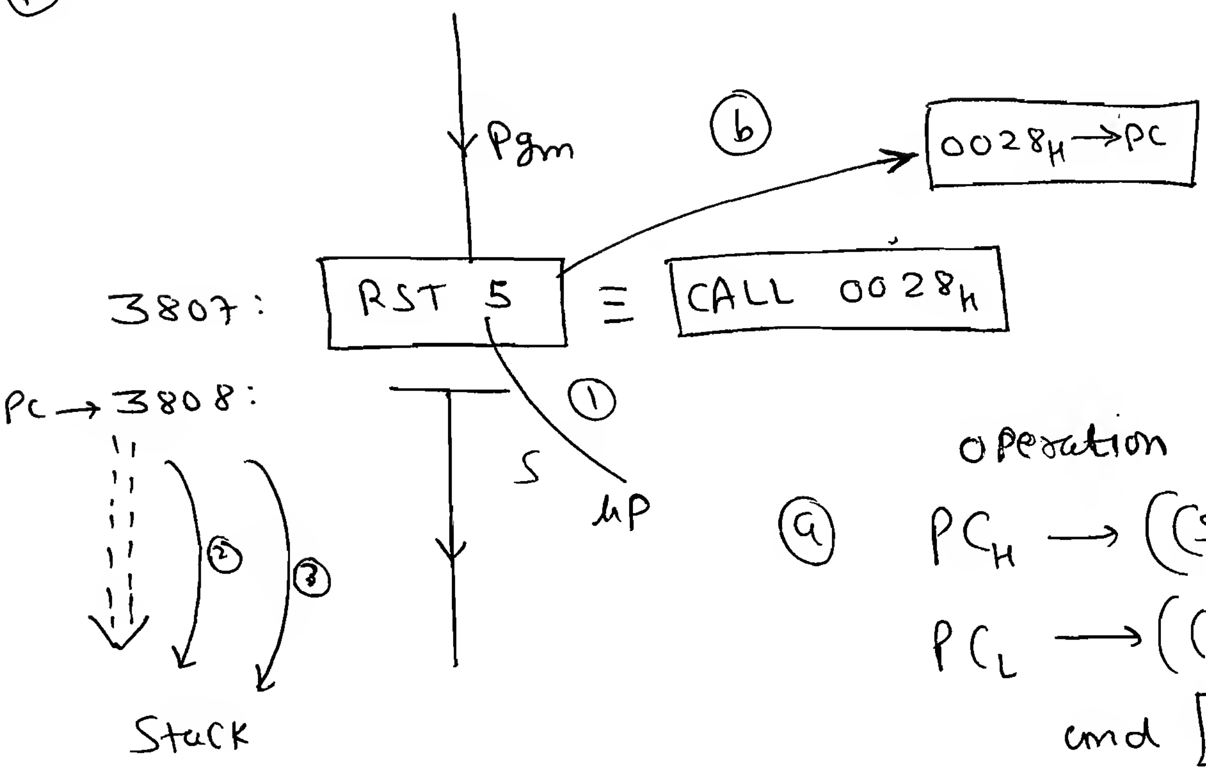
* Restart Interrupts (Software Interrupts)!!

⇒ $\boxed{RST\ n}$; $n = 0, 1, \ldots, 6, 7.$

Instruction	Opcode	Address.
RST 0	$C7_H$	0000_H
RST 1	CF_H	0008_H
RST 2	$D7_H$	0010_H
RST 3	DF_H	0018_H
RST 4	$E7_H$	0020_H
RST 5	EF_H	0028_H
RST 6	$F7_H$	0030_H
RST 7	FF_H	0038_H

Steps.

(*)



operation

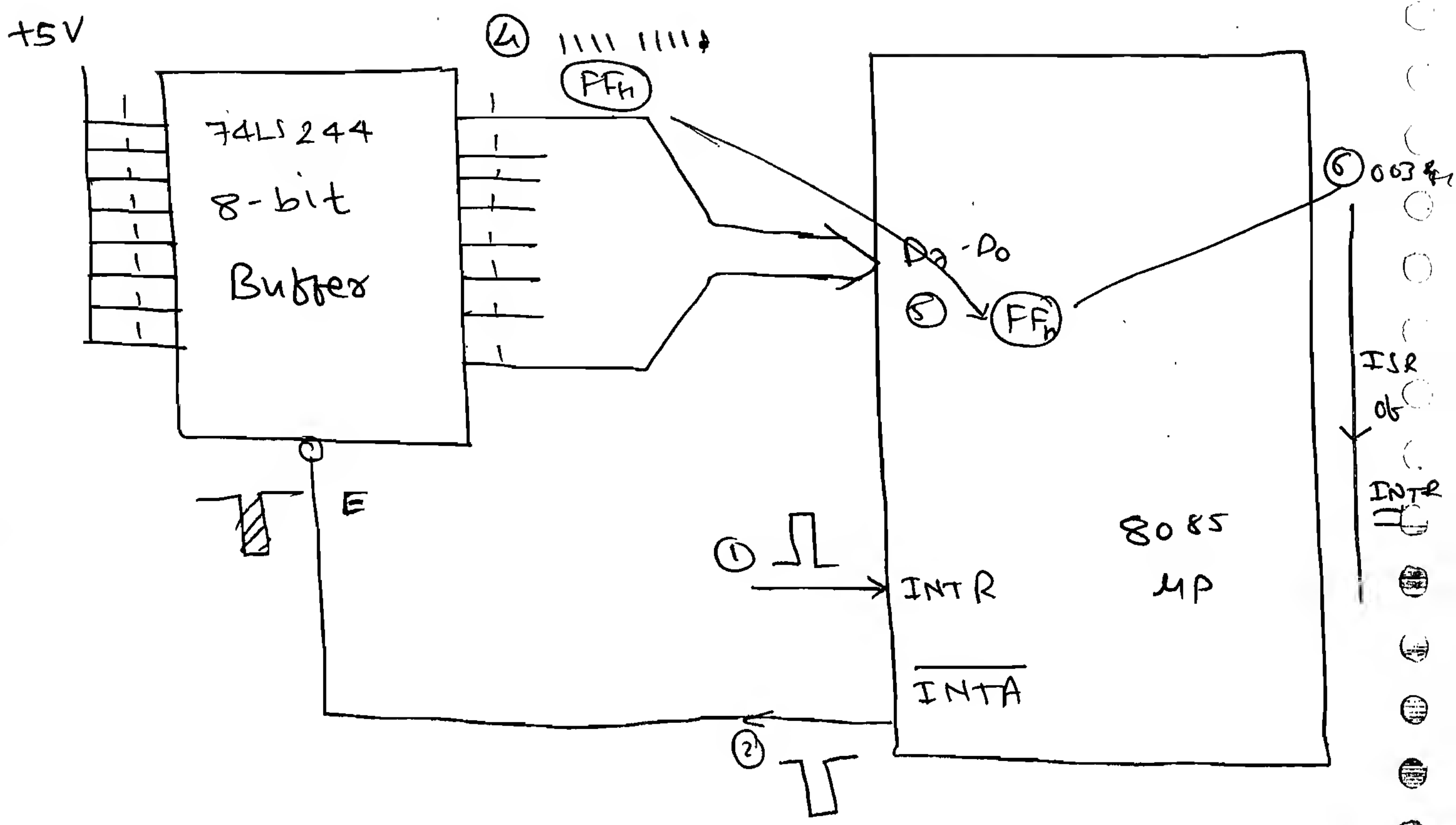
(a) $PC_H \longrightarrow ((SP-1))$

$PC_L \longrightarrow ((SP-2))$

and $\boxed{SP = SP-2.}$

(b) $(n \times 8)_H \longrightarrow PC.$

⇒ Providing Addr. Location to NVI . INTR:



Ex-1 How many times the following loops will be executed? Assume initially All the flags are cleared.

(a) LXI B, 0002_H Given $Z=0$.

Loop : DCX B
Ⓣ
$\boxed{Z=0}$
JNZ LOOP
Ⓕ
$Z=1$

$BC=0001$	$BC=0000_H$	$BC=FFFF_H$
$Z=0$ Ⓣ	$Z=$ ~~1~~ 0	$Z=0$ Ⓣ

...

* Infinite loop [∵ only Arith Instn "DCX B" doesn't affect any flags].

* $\boxed{\text{Exceptions:}}$

① ANA, ORA, XRA } ⟹ $\boxed{CY=0}$.

② INX Rp, DCX Rp } ⟹ No flags are affected.

③ INR R|M, DCR R|M } ⟹ 'CY' flag is not affected.

b) XRA A given z=0 ⟹ A = 00H. ⟹ $\boxed{Z=1}$

Imp *** LXI B, 0002 ⟹ B = 0002H

Loop: DCX B ⟹ B = 0001H.

(T) z=0 JNZ Loop. ⟶ $\boxed{Z=1}$.

↓ (F) z=1 (F) ⇓

* only once it is executed.

[∵ XRA A will set the zero flag.].

Ex-2: In the following programms How many times the decrement c instn is executed.

a) MVI C, 00H. ⟹ C = 00H

Loop: $\boxed{\text{DCR C}}$

z=0 (T) JNZ Loop. * Z=1 (F)

C = FFH = 255_{10}

C = FEH = 254_{10}

:

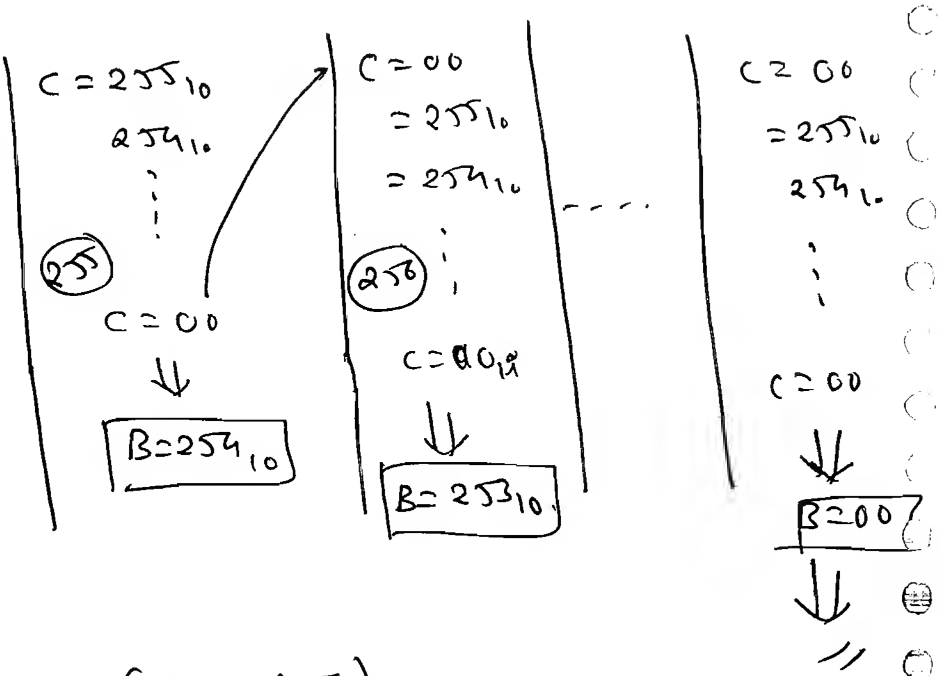
C = 00H = 0_{10} (z=1) ⟹

* $\boxed{DCR\ C}$ is executed 256 times.

b)

```
      MVI   B, 255D.  } B = C = 255D
      MVI   C, 255D.

Loop: DCR   C  ~~Loop~~
      JNZ   Loop.
      DCR   B.
      JNZ   loop.
```



Ans: (1×255) + (254 ×256) times.

DCR is executed

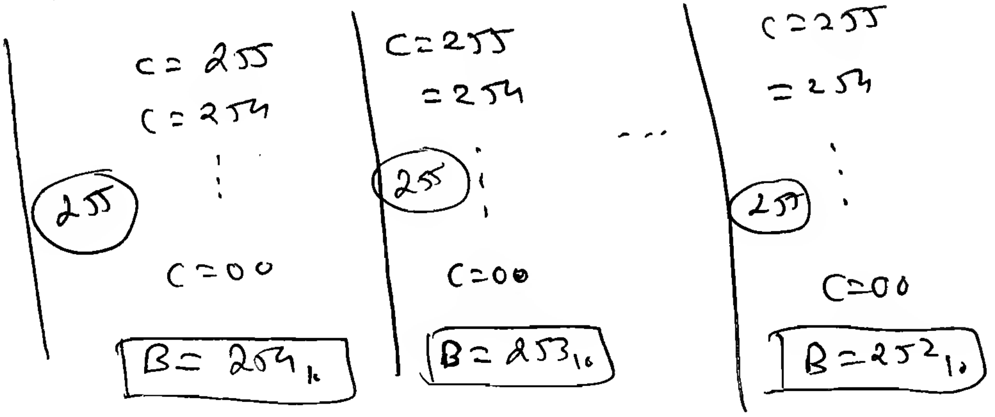
* 1st time => 255 times.

* Remaining 254 times => 256 times

Hence, $\boxed{(1 \times 255) + (254 \times 256)}$

c)

```
       MVI   B, 255D  } B=C = 255D
Loop2: MVI   C, 255D.
Loop1: DCR   C.
       JNZ   Loop1
       DCR   B
       JNZ   Loop 2
```



=> DCR is executed $\boxed{255 \times 255 \text{ times}}$

☆ Addressing Modes:

=> It is the method of specifying the operand in an instruction.

=> Instruction : $\boxed{\text{Opcode} \quad \text{Operand.}}$

Function ?

Data ?

Regs | Rp

mem

Immidiate data.

Eg. : Mov C, B

opcode operand

*** 5 Types of addressing modes:**

1. Register
2. Direct
3. Register Indirect
4. Immidiate
5. Impiicit | Impiied.

⊛

① Register Addressing Mode:

⇒ Regs | Rp

Eg. (i) MOV C, A

→ A → C

(ii) ADD R

→ A + R → R

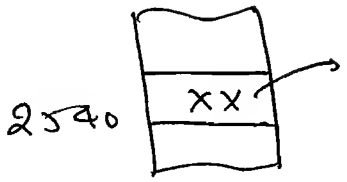
(iii) DAD rp

→ HL + rp → HL.

② Direct Addressing Mode:

→ It involes memory excess

e.g: (i) LDA 2540 ⇒



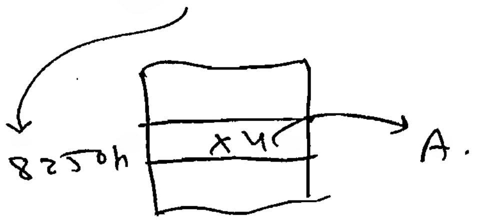
A ⇒ Direct.

→ memory Address is given directly.

③ Register Indirect Addressing mode:

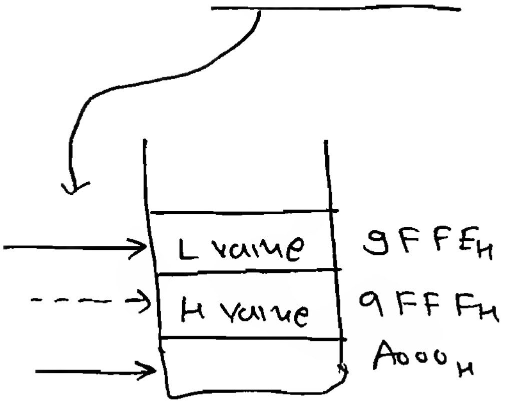
→ memory Address is specified by Rp. & R.

e.g.: (i) LDAX B ⇒ BC = 8250H



So, Reg indirect.

(ii) Push H ⇒ SP = $A000_H$. ⇒ Reg. Indirect



④ Immediate Addressing mode:

⇒ Source operand is alway 8 bit | 16 bit value.

E.g. : (i) MVI C, 32 ⇒ $\boxed{32 \rightarrow C}$

(ii) ADI 45 ⇒ $\boxed{A + 45 \rightarrow A.}$

(iii) LXI B, 1020_H ⇒ $\boxed{1020_H \longrightarrow BC}$

(iv) XRI 30_H ⇒ A ⊕ 30 ⟶ A.

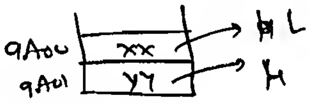
⑤ Implicit | Implied Addressing mode:

⇒ Source and destination are predefine.

E.g. RAL, RAR, CMA, CMC, STC, PCHL,

XCHG, PCHL, XTHL..

Ex-1 Determine the addressing modes of the following instructions:



1) LHLD 9A00. ⟶ Direct

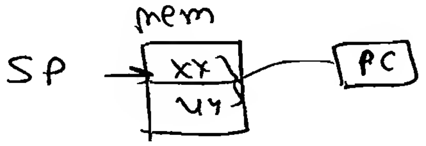
2) MOV C, M ⟶ Reg. Indirect

3) NOP/HLT ⟶ ~~Implicit~~ & ~~Implied~~ No Addr. mode.

4) JMP 8250 ⟶ $\boxed{8250 \rightarrow PC}$ ⟶ Immediate

5) MVI M, 32. => Reg. Indirect and also Immediate.

✓ 6) RET => Reg. Indirect.



=> Reg. Indirect.

✓ 7) CALL 3800 => Reg. Indirect / Immediate.

8) DAA => Implied.

P) Which of the following Insth is getting executed given ~~that~~ following:

Address : 4040H

Data : 44H

IOmapped I/O ← IO/$\overline{M}$: 1

X $\overline{RD}$: 1

writing ← $\overline{WR}$: 0.

(a) IN 40H

(b) IN 44H.

✓ (c) OUT 40H

(d) OUT 44H

P) Which of the following will store the value of accumulator into a output device, whose address is $\boxed{F250_H}$ → mem. mapped I/O

i) LXI H, F250
mov A, M

(ii) LXI H, F250
OUT M

(iii) LXI H, F250
OUT F250

✓ (iv) LXI H, F250.
mov M, A.
→ A ⟶ M
A → i.e. output device.

* Delay Program:

T-States

1) MVI B, FF — 7T

Loop: DCR B — 4T ← False

JNZ Loop. — 7T / 10T ← True

≈10T

$\Rightarrow \quad T_D = T_0 + T_L;$

T_0 = Delay outside Loop.

$T_0 = 7T = (7 \times 320ns)$

T_L = Delay within the loop.

= (Count$_{10}$ × No. of T-states in loop × T)

$T_L = (255_{10} \times 14 \times 320ns)$

$$\therefore \quad T_D = (7 \times 320ns) + (255 \times 14 \times 320ns) - \underbrace{(3 \times 320ns)}_{\text{correction.}}$$

2) LXI B, FFFF$_h$ — 10T

Loop: DCX B — 6T

MOV A, C — 4T

ORA B — 4T

JNZ LOOP — 10T / 7T

Z=1 — True, False

⇒ $T_0 = 10T = [10 \times 320ns]$

$T_L = [Count_{10} \times 24T]$

$= [65,535 \times 24 \times 320ns]$

∴ $T_D = T_0 + T_L$

$T_D = [10 \times 320ns] + [65,535 \times 24 \times 320ns]$.

* Nested loop: (Loop within loop).

```
                                                     T-states.
⇒        MVI   B,  Count 2                             7T
Loop 2:  MVI   C,  Count 1                             7T
Loop 1:  DCR   C          ]                            4T
         JNZ   Loop 1     ] T_L1                      ≈10T
         DCR   B                                       4T
         JNZ   Loop 2                                 ≈10T
              ⇓
```

$$\boxed{T_D = [21T + T_{L_1}]\, count2_{10} + [7 \times 320ns] - [Count2 \times 3T] - (3T].}$$

Where $T_{L_1} = [Count1_{10} \times 14 \times 320ns]$.

* $T_D = (21T + T_{L_1})\, Count2_{10} \overset{+}{\cancel{\times}} (7 \times 320ns) - [Count2 + 1]\, 3T.$

→ JNZ Loop 1 is false Count 2 times.

⇒ Subtract [count 2 × 3T].

JNZ loop 2 is false only one time.

⇒ Subtract 3T.

* General Points:

① ⓐ 1-Byte unconditional CALL Instruction = $\boxed{\text{RSTn}}$

ⓑ 1-Byte unconditional JMP Instruction = $\boxed{\text{PCHL}}$

② The DMA input HOLD is having the preference over non-maskable interrupt TRAP.

③ µp Checks the ready pin in 'T₂' state of each machine cycle. If it finds it as 'zero' it inserts wait T-states betn T_2 & T_3 untill the ready becomes one.

④ How to shift 16 bit data towards left by 1 bit?

⇒ i) Store 16 bit no. into HL pair.

ii) Use $\boxed{\text{DAD H}}$ instruction.

$$\begin{array}{lcccccc} & & & \leftarrow & & & \\ HL = & 0000 & 0000 & 0000 & 0100_2 & = & 4_{10} \\ +HL = & 0000 & 0000 & 0000 & 1000_2 & = & \times 2 \\ \hline \underline{\underline{HL}} & & & & & & = 8_{10} \end{array}$$

⑤ In I/O mapped I/O mode can we give the same address to a input device and o/p device?

⇒ Yes, because they can be recognized based on their control signals. Hence, in this mode we can connect 256 input devices and 256 output devices in total 512.

Input Device	Output Device
Addr = F4H	Addr = F4H

8085 μP

① Address ⇒ $\boxed{A_7 - A_0}$ = F4H | $A_{15} - A_8$ = F4H

② Control signal ⇒ $IO/\overline{M} = 1$

$\overline{WR} = 0$ → output device.

③ Data